Epistemic Specifications and Conformant Planning

Yan Zhang
School of Computing, Engineering and Mathematics,
University of Western Sydney, Australia
Yan.Zhang@westernsydney.edu.au

Yuanlin Zhang
Department of Computer Science,
Texas Tech University, USA
y.zhang@ttu.edu

Abstract

Epistemic Specifications allow for the correct representation of incomplete information in the presence of multiple belief sets by expanding Answer Set Programming with modal operators K and M. The meaning of M in the existing work does not correspond well to the principle of justifiedness accepted by the community. It is, however, challenging to characterize the justfiedness of each belief, due to the complexity introduced by M. We address this issue by identifying a belief set with a program which uniquely decides the belief set. This idea leads to a novel definition of the semantics of Epistemic Specifications which assures that each belief in any belief set is well justified. We also show that conformant planning problems can be naturally represented by Epistemic Specification under our semantics.

Introduction

Answer Set Programming (ASP) is currently a dominant logic based representation paradigm in the knowledge representation community and has found numerous applications (Gelfond and Kahl 2014; Erdem, Lee, and Lierler 2012). Epistemic Specifications (Gelfond 1994) extend ASP by allowing introspective reasoning, i.e., reasoning with multiple belief sets, through the use of modal operators K and M. There are many potential applications of Epistemic Specifications including conformant planning, autonomous robot control and policy management (Kahl et al. 2015).

Since Gelfond first proposed a language of Epistemic Specifications (Gelfond 1994), several alternative approaches have been developed in recent years (del Cerro, Herzig, and Su 2015; Gelfond 2011; Kahl 2014; Kahl et al. 2015; Truszczynski 2011; Wang and Zhang 2005; Shen and Eiter 2016). One persistent challenge driving these works is how to address *circular justification*, which is generally thought of as an undesirable property, associated with the M operator. Consider a simple program Π_1 consisting of one rule:

$$p \leftarrow Mp.$$

which is read as "if it is possible for an agent to believe p, the agent should believe p." All existing semantics allow $\{\{p\}\}$ to be a world view of Π_1. In this world view, there is a strong sense of circular justification for p. To support existing semantics, a third party provides the following example:

$$flies(X) \leftarrow bird(X), Mflies(X).$$

With a set of facts on `bird`, the program concludes that all birds fly, which is taken as natural and intended. However, a similar program

$$flies(X) \leftarrow horse(X), Mflies(X),$$

with a set of facts on `horse`, will conclude, by existing semantics, that all horses fly, which does not seem to be natural or intended. This example is certainly not to show that the existing approach is wrong, but we feel that much work is needed to have a better understanding of M operator.

The aim of this paper is to develop an intuitive understanding of M operator and a semantics to get rid of circular justification in a stronger sense. Under this new semantics, the world view of Π_1 is $\{\{\}\}$. (Gelfond 2011; Kahl 2014; del Cerro, Herzig, and Su 2015; Shen and Eiter 2016) have addressed circular justification to various extent but all in a weaker sense than ours since they all accept $\{\{p\}\}$ as a world view for Π_1.

We follow the *rationality principle* (Gelfond and Kahl 2014) which says that *a rational agent should believe only what he is forced to believe*, and we understand Mp, i.e., p is possible, as that *belief in p is forced in some belief set of the rational agent*. To formalize this intuition in the context of Epistemic Specifications, the classical techniques, such as reduct based or fixpoint based, in defining ASP semantics do not lend themselves immediately to it. We resort to a new reduct approach to capture our intended semantics.

We illustrate our ideas by considering a program Π_2:

$r_1 : p \leftarrow Mq, not\ q,$
$r_2 : q \leftarrow Mp, not\ p.$

Let $W = \{A_1, A_2\}$ where $A_1 = \{p\}$ and $A_2 = \{q\}$. Existing approaches accept W as a world view. It seems that p is forced (by rule r_1) in A_1 due to q in A_2, and vice versa. For us, the belief p and q in different belief sets form a circular justification.

To address circular justification, *for each belief set A of a world view W, we construct an ASP program from A, W and the Epistemic Specification to justify the beliefs of A.* For example, first consider A_1 and $W = \{A_1, A_2\}$. All atoms in Π_2, without M before them, can be understood

in terms of A_1 while atoms prefixed by M can be understood in terms of W. Hence, we add a subscript of 1 to all atoms without M before them to denote that they are used to specify A_1. As for Mq in rule r_1, it is understood as a q in some belief set of W, and it can only be the q of A_2. r_1 is understood as (note q_2 below refers to q in A_2) r_1': $p_1 \leftarrow q_2, not\ q_1$. Note we don't remove Mq in this understanding because q_2 may in turn depend on beliefs in A_1. By replacing Mq by q_2, circular justification can be taken care of by the resulting regular logic program(s). Since $q \notin A_1$, we can remove $not\ q_1$ as in classical ASP reduct (remember that q_1 denotes that the satisfaction of q depends on A_1). As a result, we obtain, from r_1' and A_1, the rule r_{11}: $p_1 \leftarrow q_2$. For r_2, Mp in its body can be only understood, by W, as $p \in A_1$. However, since $p \in A_1$, $not\ p_1$ is not satisfied by A_1. Hence, the rule is useless to justify any beliefs of A_1. So, r_{11} is the only rule justifying A_1 (wrt W). We use Π_{21} to denote the rule.

In the same manner, when considering A_2 and W, we obtain the program Π_{22} with one rule r_{22}: $q_2 \leftarrow p_1$.

Since Π_{21} refers to an atom in Π_{22} which in turn refers to an atom in Π_{21}, we may not use Π_{21} itself to justify A_1. Instead, we should use $\Pi_{21} \cup \Pi_{22}$ to justify A_1 and A_2. There is a clear circular justification in $\Pi_{21} \cup \Pi_{22}$, which is automatically eliminated by the answer set of $\Pi_{21} \cup \Pi_{22}$. A_1 and A_2 are not justified by $\Pi_{21} \cup \Pi_{22}$. So, W is not an intended world view. In fact, $\{\{\}\}$ is the only collection that can be justified in terms of the approach above.

The main contributions of this paper are of twofold:

- With the help of a new reduct technique, we propose a new formal semantics for Epistemic Specifications. The semantics prevents circular justifications in a sense stronger than the existing approaches.
- We also present a natural and elaboration tolerant representation of a class of conformant planning problems, which provides new insights for the representation of planning problems with epistemic reasoning capacity.

The rest of this paper is organized as follows. Section 2 introduces necessary notions and basic definitions that will be used throughout the paper. Section 3 proposes a new reduct for disjunctive programs which is essential for defining the new semantics of epistemic specifications. Section 4 presents the definition of justified views and explains how this new semantics overcomes the difficulties of circular justification that previous approaches suffer from. Then section 5 provides a revision on justified view definition by taking maximality into account, which eventually avoids some unintuitive results. Section 6 demonstrates a natural and elaboration tolerant representation of a class of conformant planning problems using our new semantics. Finally, section 7 concludes this paper with some remarks.

Preliminaries

We consider a language $\mathcal{L}$ of traditional propositional answer set programs expanded with two modal operators named K and M. *Atoms and literals* are defined as usual, while we also call them *objective* atoms or literals. For a given objective atom (literal) l, Kl and Ml are called *subjective atoms (literals)*. Intuitively, we may read Kl as "l is known to be true", and Ml is read as "l may be believed to be true". We also allow classical negation $\neg$ to appear in front of a subjective literal, such as $\neg Kl$ and $\neg Ml$. Then we simply call Kl, Ml, $\neg Kl$ and $\neg Ml$ *extended subjective literals*. For simplicity, in the following we may also use αl to denote an extended subjective literal, where α is K, M, $\neg K$ or $\neg M$.

A *belief set* is a set of objective literals. A *view* is defined to be a collection of belief sets. The *satisfaction* of an extended subjective literal is then defined based on a view. Let $W = \{A_1, \cdots, A_k\}$ be a view. We define the satisfaction of an extended subjective literal in W as follows:

- $W \models Kl$ iff $\forall A_i \in W, l \in A_i$;
- $W \models Ml$ iff $\exists A_i \in W, l \in A_i$;
- $W \models \neg Kl$ iff $W \not\models Kl$;
- $W \models \neg Ml$ iff $W \not\models Ml$.

Now we specify an *Epistemic Specification (ES) program* to be a finite set of rules of the form:

$$l_1 \vee \cdots \vee l_k \leftarrow l_{k+1}, \cdots, l_m, not\ l_{m+1}, \cdots, not\ l_n, \quad (1)$$

where $l_1, \cdots, l_k$ and $l_{m+1}, \cdots, l_n$ are objective literals, and $l_1, \cdots, l_m$ are either objective literals or extended subjective literals. Here we also call "$not\ l$" a *weak negated literal*.

Let A be a belief set and l an objective literal. We say that l *is satisfied* in A, denoted as $A \models l$, if $l \in A$; while $not\ l$ *is satisfied* in A, denoted as $A \models not\ l$, if $l \notin A$. A disjunction of literals $l_1 \vee \cdots \vee l_k$ *is satisfied* in A, denoted as $A \models l_1 \vee \cdots \vee l_k$, if for some l_i ($1 \leq i \leq k$), $l_i \in A$.

Now let r be a rule of the form (1) and W a view. We say that r *is satisfied* in W, denoted by $W \models r$, if for each extended subjective literal l and each objective literal l' in $\{l_{k+1}, \cdots, l_m\}$, $W \models l$, $A \models l'$ for all $A \in W$, and for each $l'' \in \{l_{m+1}, \cdots, l_n\}$, $A \models not\ l''$ for all $A \in W$, then $A \models l_1 \vee \cdots \vee l_k$ for all $A \in W$. Sometimes, we use "$head(r) \leftarrow body(r)$" to represent r's form (1). Also, if r's body is empty, we simply represent (1) as "$head(r)$", while when "$head(r)$" is empty, we call "$\leftarrow body(r)$" a *constraint*.

An ES program Π is called *positive* if for each rule r in Π, r does not contain any weak negated literals. Also, if Π does not contain any extended subjective literals, Π is reduced to a traditional disjunctive extended logic program, or simply called a *disjunctive program*.

Given a disjunctive program Π and a belief set A, we say that a rule r in Π *is satified* in A if $A \models body(r)$ implies $A \models head(r)$. Π *is satisfied* in A, or A is a *model* of Π, if each rule in Π is satisfied in A. The *reduct* of Π wrt A, denoted by Π^A, is the program obtained from Π by first removing rules whose body contains $not\ l$ such that $l \in A$ and then removing all weak negated literals. A is an *answer set* of Π if A is a minimal model of Π^A.

Disjunction Reduct

Our ideas illustrated in section 1 might not work when the program constructed for a belief set A from a view W has

more than one answer set. The disjunction in the head of a rule may result in multiple answer sets of a program. To overcome this, here we introduce the disjunction reduct of a disjunctive program with respect to a belief set. This reduct ensures a unique answer set so that the reduct can be taken as the support/justification of the belief set when its answer set coincides with the belief set.

Definition 1 (Disjunction reduct) *The* disjunction reduct *of a positive disjunctive program* Π *wrt a belief set* A*, denoted by* $\Pi^{A,\vee}$*, is a program resulting from* Π *by removing all literals not in* A *from the head of all the rules of* Π*.*

Example 1 Consider program $\Pi_3 = \{p \vee q\}$. Given $A_1 = \{p\}$ and $A_2 = \{q\}$, then we have $\Pi_3^{A_1,\vee} = \{q\}$ and $\Pi_3^{A_2,\vee} = \{p\}$ respectively. □

The intuition behind the disjunction reduct is quite clear: if a literal occurring in the head of a rule is not contained in the underlying belief set, then this literal is not derivable wrt this belief set, and hence does not affect other literals in the head. So we simply remove it from the head. The following result, whose proof is in the supplement, shows a close connection between the disjunction reduct and the classical semantics of disjunctive programs.

Proposition 1 *A is an answer of a positive disjunctive program Π iff A satisfies all rules of Π and A is the answer set of the disjunction reduct of Π wrt A.*

Justified Views

In this section, we give a precise definition on justified views for ES programs. Firstly, as illustrated in the introduction section, we need to define a reading/interpretation on all extended subjective literals occurring in an ES program against every belief set in the given view, and such interpretation provides a key step in defining a justified view.

Definition 2 (Modal operator interpretation) *Consider an ES program Π and a collection W of belief sets $\{A_1, ..., A_n\}$. Let EL_Π be the set of all occurrences of extended subjective literals in Π, and $OL_\Pi = \{l_i, not\ l_i \mid l$ is an objective literal occurring in Π and $i \in 1..n\}$. A* modal operator interpretation *for Π wrt view W is a mapping ρ from $EL_\Pi \times 1..n$ to OL_Π, defined as follows:*

$$\forall i\ \rho(Kl, i) = l_i,\ \textit{if}\ W \models Kl; \tag{2}$$

$$\forall i\ \rho(Ml, i) = l_j,\ \textit{if}\ W \models Ml\ \textit{and}\ l \in A_j; \tag{3}$$

$$\forall i\ \rho(\neg Kl, i) = not\ l_j,\ \textit{if}\ W \models \neg Kl\ \textit{and}\ l \notin A_j; \tag{4}$$

$$\forall i\ \rho(\neg Ml, i) = not\ l_i,\ \textit{if}\ W \models \neg Ml. \tag{5}$$

Let us take a closer look at Definition 2. Basically, mapping ρ provides an interpretation on an extended subjective literal against some particular belief set in the given view. For instance, if subjective literal Kl is satisfied in view W, then for every belief set $A_i \in W$ $(1 \leq i \leq n)$, l must be in A_i. In this case, we interpret Kl as l_i, indicating that the objective literal l is in belief set A_i. This interpretation is specified by (2). On the other hand, suppose the subjective literal Ml is satisfied in W. Then according to the semantics, there exists some A_j such that $l \in A_j$, for which we view as a supporting evidence for Ml being satisfied in W and assign $\rho(Ml, i) = l_j$, as depicted in (3). The negative subjective literals such as $\neg Kl$ and $\neg Ml$ are specified by (4) and (5), respectively, based on similar explanations.

It should be noted that we do not provide a modal operator interpretation for an extended subjective literal αl if it is not satisfied in W. This is because a rule containing an unsatisfied extended subjective literal will be simply removed during the process of generating a modal reduct, as will be shown later.

Example 2 Consider program Π_4 as follows:

$p \leftarrow Mq, not\ q$,
$q \leftarrow Mp, not\ p$,
$\leftarrow not\ p, not\ q$.

Let $A_1 = \{p\}$, $A_2 = \{q\}$ and $W = \{A_1, A_2\}$. The only modal operator interpretation wrt W is $\rho(Mp, i) = p_1$ and $\rho(Mq, i) = q_2$ for all $i \in 1..2$. □

Definition 3 (Modal reduct) *Consider an ES program Π, a collection W of belief sets $\{A_1, ..., A_n\}$, and a modal operator interpretation ρ with respect to W and belief set A_i. The* modal reduct *of Π based on W, A_i and ρ, denoted as $\Pi^{W,A_i,\rho}$, is the program obtained from Π by the following three steps:*

(1) renaming each literal l not occurring in any subjective literal in Π by l_i;

(2) removing rules whose body contains αl such that $W \not\models \alpha l$, and finally,

(3) replacing every occurrence of extended subjective literal αl in the remaining program by $\rho(\alpha l, i)$.

Intuitively, by generating the modal reduct, we reduce the ES program Π to a disjunctive logic program $\Pi^{W,A_i,\rho}$, in which all objective literals in Π not occurring in any subjective literals are labelled with subscript i indicating that they are explicitly associated with belief set A_i. Furthermore, all rules containing not satisfied extended subjective literals in W will be removed, and all other satisfied extended subjective literals in W are then replaced by their corresponding objective or weak negated objective literals with proper justifications under ρ. The following example illustrates more details about this transformation.

Example 3 [Example 2 continued] Still consider program Π_4 as in Example 2. We consider a collection $W = \{A_1, A_2\}$ of belief sets, where $A_1 = \{p\}$, $A_2 = \{q\}$. Then it is easy to see that $\rho(Mq, i) = q_2$ and $\rho(Mp, i) = p_1$ $(i = 1, 2)$ is a modal operator interpretation. Then we have the following two modal reducts: $\Pi_4^{W,A_1,\rho} = \{\, p_1 \leftarrow q_2, not\ q_1.\ q_1 \leftarrow p_1, not\ p_1.\ \leftarrow not\ p_1, not\ q_1\}$, and $\Pi_4^{W,A_2,\rho} = \{\, p_2 \leftarrow q_2, not\ q_2.\ q_2 \leftarrow p_1, not\ p_2.\ \leftarrow not\ p_2, not\ q_2\}$. □

Now we are in a position to present the key definition - justified view.

Definition 4 (Justified view) *Consider an ES program Π and a collection W of belief sets $\{A_1, ..., A_n\}$. Let $B = \{l_i \mid l \in A_i, i \in 1..n\}$. A* full reduct *of Π with respect to W, A_i and a modal operator interpretation ρ, denoted by $\Pi^{W,A_i\rho,not,\vee}$, is the program obtained by applying modal reduct based on W, A_i and ρ, Gelfond-Lifschitz*

reduct and disjunction reduct with respect to B in sequence to Π: $((\Pi^{W,A_i\rho})^B)^{B,\vee}$. *$W$ is a* justified view *of Π if there exists a modal operator interpretation ρ such that B is the answer set of the program* $\bigcup_{i=1}^{n} \Pi^{W,A_i,\rho,not,\vee}$.

Example 4 [Example 3 continued] Let us consider program Π_4 in Example 2 once again. As in Example 3, let $W = \{A_1, A_2\}$, where $A_1 = \{p\}$, $A_2 = \{q\}$, and $\rho(Mq, i) = q_2$ and $\rho(Mp, i) = p_1$ $(i = 1, 2)$. Then we have $\Pi_4^{W,A_1,\rho}$ and $\Pi_4^{W,A_2,\rho}$ as showed in Example 3.

Let $B_1 = \{p_1\}$ and $B_2 = \{q_2\}$ and $B = B_1 \cup B_2$. From Definition 4, we then have $(\Pi_4^{W,A_1,\rho})^B = \{\, p_1 \leftarrow q_2\}$, and $(\Pi_4^{W,A_2,\rho})^B = \{\, q_2 \leftarrow p_1\}$. Since the heads of the rules do not have disjunctions, the programs above keep the same after disjunction reduct. Therefore, $\Pi_4^{W,A_1,\rho,not,\vee} = \{p_1 \leftarrow q_2\}$, denoted by Π_{41}, and $\Pi_4^{W,A_2,\rho,not,\vee} = \{q_2 \leftarrow p_1\}$, denoted by Π_{42}. We can see that B is not an answer set of program $\Pi_{41} \cup \Pi_{42}$. So W is not justified.

Now we consider $W' = \{A'_1\}$, where $A'_1 = \{\}$. Since $W' \not\models Mp$ and $W' \not\models Mq$, no modal interpretation is needed. So the only full reduct is: $\{\leftarrow\}$, which, clearly, has no answer set. This follows that W' is not justified either. In fact, it is not difficult to show that Π_4 does not have any justified view. □

Without getting into details, we can also show that program Π_2, which is illustrated in the introduction section and the same as Π_4 except not including the constraint "$\leftarrow not\ p, not\ q$", has a unique justified view $\{\{\}\}$, according to our previous definitions.

Example 5 Consider the program Π_5:

$p \vee q$,
$q \vee s$,
$a \leftarrow \neg Mp$,
$b \leftarrow \neg Ms$,
$c \leftarrow not\ a$,
$d \leftarrow not\ b$.

Let $W_1 = \{\{p, s, c, d\}, \{q, c, d\}\}$ and $W_2 = \{\{q, a, b\}\}$. Since $W_1 \not\models \neg Mp$ and $W_1 \not\models \neg Ms$, there is "no" modal operator interpretation for extended subjective literals in Π_4 under W_1; while $\rho(\neg Mp, 1)) = not\ p_1$ and $\rho(\neg Ms, 1)) = not\ s_1$ provide modal operator interpretations under W_2. We also have $B = \{p_1, s_1, c_1, d_1, q_2, c_2, d_2\}$ and $B' = \{q_1, a_1, b_1\}$ based on W_1 and W_2, respectively.

For the case of W_1, we obtain two full reducts, denoted by Π_{51} and Π_{52} respectively, as follows: $\Pi_{51} = \{\, p_1.\ s_1.\ c_1.\ d_1\}$, and $\Pi_{52} = \{\, q_2.\ c_2.\ d_2\}$. Clearly, the answer set of $\Pi_{51} \cup \Pi_{52} = B = \{p_1, s_1, c_1, d_1, q_2, c_2, d_2\}$. So W_1 is a justified view of Π_5. On the other hand, for the case of W_2, there is one full reduct: $\{q_1.\ a_1.\ b_1\}$ whose answer set is $B' = \{q_1, a_1, b_1\}$. So W_2 is also a justified view of Π_5.

In fact, we can further show that W_1 and W_2 are the only two justified views of program Π_5. □

World views: Integrating Justifiedness and maximality

As we have shown in the last section, justified views provide a sound basis for defining the semantics of ES programs. Now the question is: may we use the justified view as the final semantics for ES programs? Let us first consider a simple program Π_6 consisting of a single rule:

$p \vee q$.

It is not hard to see that Π_6 has three justified views: $\{\{p\}\}$, $\{\{q\}\}$ and $\{\{p\}, \{q\}\}$. Obviously only the last one should be a rational model for Π_6.

What the justified view lacks is the maximality that we should capture for reasoning about incomplete information. In this section, we integrate such maximality into our justified views and therefore to provide a semantic foundation for ES programs.

Definition 5 (Maximal view) *Let Π be an ES program and $W = \{A_1, \cdots, A_n\}$, where $A_1, \cdots, A_n$ are belief sets. A disjunctive program Π^W is called the* general modal reduct *of Π with respect to W, denoted by Π^W, if it is obtained from Π by performing the transformation for every rule $r \in \Pi$:*

(1) for every Kl occurring in r, replacing it by l if $W \models Kl$, otherwise removing r from Π;

(2) for every Ml occurring in r, removing it from r's body if $W \models Ml$, otherwise removing r from Π;

(3) for every $\neg Kl$ occurring in r, removing it from r's body if $W \models \neg Kl$, otherwise removing r from Π;

(4) for every $\neg Ml$ occurring in r, replacing it by $not\ l$ if $W \models \neg Ml$, otherwise removing r from Π.

We call W a maximal view *if W is the collection of all answer sets of Π^W.*

It is worth mentioning the difference between the modal reduct defined in Definition 3 and the general modal reduct defined above. In the former transformation, an extended subjective literal αl is either replaced by its modal operator interpretation explicitly associating to its belief set justification, or causes an elimination of the rule if it is not satisfied in the underlying view.

The general modal reduct, on the other hand, is to maximally retain the objective literal information during the process of eliminating extended subjective literals. For instance, consider condition (4) in Definition 5, if $W \models \neg Ml$, it implies that for each $A \in W$, $l \notin A$, in this case, instead of removing it from r's body, which is equivalently to replace it by $\mathbf{T}$, we replace $\neg Ml$ by $not\ l$ to keep objective literal information in the resulting rule. Note that for condition (3), we indeed remove $\neg Kl$ from r's body if $W \models \neg Kl$. This is because in this case although we know that there exists some belief set not containing objective literal l, we do not know exactly which belief set. So conservatively, we simply assume $\neg Kl$ to be true.

Based on Definition 5, it is simple to check that $\{\{p\}, \{q\}\}$ is the unique maximal view of Π_6. It is also easy to see that not all maximal views are justified. For instance, for program $\Pi_1 = \{p \leftarrow Mp\}$ mentioned in Introduction,

both $\{\{\}\}$ and $\{\{p\}\}$ are maximal, but only the first one is justified.

Definition 6 (World view) *Let Π be an ES program and W a collection of belief sets. W is a* world view *of Π iff W is a justified and maximal view of Π.*

Example 6 Consider Program Π_7:

$p \vee q,$
$q \vee r,$
$p \leftarrow Mp,$
$s \leftarrow p, q,$
$s \leftarrow Ms,$
$\leftarrow p, not\ s.$

Let $W_1 = \{\{p, q, s\}, \{p, r, s\}\}$ and $W_2 = \{\{q\}\}$. It can be showed that both W_1 and W_2 are maximal.

From Definitions 2 and 3, we can also obtain the two modal reducts wrt to $\{p, q, s\}$ and $\{p, r, s\}$ as follows: $\{p_1 \vee q_1.\ q_1.\ p_1 \leftarrow p_2.\ s_1 \leftarrow p_1, q_1.\ s_1 \leftarrow s_1.\ \leftarrow p_1, not\ s_1\}$, and $\{p_2.\ r_2.\ p_2 \leftarrow p_2.\ s_2 \leftarrow p_2, q_2.\ s_2 \leftarrow s_1.\ \leftarrow p_2, not\ s_2\}$. We can verify that W_1 is justified according to Definition 4. Similarly, we can also verify that W_2 is justified. Therefore, W_1 and W_2 are two world views of Π_7. □

An Application in Conformant Planning

In this section, we illustrate an application of our ES programs for conformant planning. Our definition of conformant planning is based on the action language $\mathcal{AL}$ (Turner 1997; Baral and Gelfond 2000) and the work in (Tu et al. 2011).

Space limitation does not allow us to include definitions of $\mathcal{AL}$ *statements* (*causal laws, executibility conditions* and *state constraints*), *fluents*, *actions*, *states* and *transition diagram* here. These definitions can be found in (Gelfond and Kahl 2014). A *system description* is a set of $\mathcal{AL}$ statements. It is used to specify a transition diagram of a dynamic domain. Given a system description D, we use $T(D)$ to denote the transition diagram specified by D. We use $\Pi(D, \sigma, \{a\})$, where D is a system description, σ a state and a a set of actions, to denote the ASP program $\Pi(D) \cup \{holds(l, 0) : l \in \sigma\} \cup \{occurs(a_i, 0) : a_i \in a\}$ where $\Pi(D)$ is an ASP program obtained from D as in (Gelfond and Kahl 2014). Actions of a are said to be *prohibited* in a state σ of a transition diagram $T(D)$ defined by a system description D if D contains an executibility condition for actions $a_0 \subseteq a$ whose body is satisfied by σ; otherwise, actions in a are said to be *executable* in σ. A sequence of actions $a_0, \ldots, a_{n-1}$ is *executable* in a state σ if the sequence is empty or a_0 is executable in σ and the sequence $a_1, \ldots, a_{n-1}$ is executable in σ' for every $\langle \sigma, a_0, \sigma' \rangle \in T(D)$.

A system description D is *consistent* if for any state σ_1 and an action a executable in σ_1, there exists at least one state σ_2 such that $\langle \sigma_1, a, \sigma_2 \rangle \in T(D)$. A system description is *deterministic* if for any state σ_1 and action a, there is at most one state σ_2 such that $\langle \sigma_1, a, \sigma_2 \rangle$ is a transition defined by $T(D)$. A system description D is *stable* if for any state σ, $\Pi(D, \sigma, \{\})$ has a unique answer set A and $\sigma = \{l : holds(l, 1) \in A\}$. Intuitively, a system description is stable if no matter what state the system is in, the system keeps in the same state when no action occurs.

A *conformant planning problem* is a triple $\langle D, \Sigma, g \rangle$ which consists of a system description D of an action language $\mathcal{AL}$, a collection of the possible initial states Σ, and a goal g where g is a set of fluent literals. A sequence $\alpha = \langle a_0, \ldots, a_{n-1} \rangle$ of actions is called a *solution* to $\mathcal{P}$ if α is executable in every state of Σ and for any path $\sigma_0, a_0, \ldots, a_{n-1}, \sigma_n$ of $T(D)$ where $\sigma_0 \in \Sigma$, g is true in σ_n, i.e., $g \subseteq \sigma_n$. A solution α of a conformant planning problem P is *simple* if no proper prefix of α is a solution of P.

Given a conformant planning problem $P = \langle D, \Sigma, g \rangle$, let m be the maximal number of steps allowed. We construct the following ES program, denoted by $\tau_m(P)$, whose world views contain solutions of the problem.

1. Rules for the dynamic domain. For each statement of D, translate it to ES rule(s), as defined in (Gelfond and Kahl 2014), except the statements of executability condition which are translated as follows: for each executability condition

   $$\textbf{impossible}\ a\ \textbf{if}\ l_1, \ldots, l_n,$$

   it is translated into

   $$prohibited(a, S) \leftarrow holds(l_1, S), \ldots, holds(l_n, S)$$

   where $prohibited$ is a new predicate, $prohibited(a, S)$ denotes action a is prohibited at step S and $holds(l, S)$ denotes that fluent l holds at step S. Finally include the rules for inertia axioms over non defined fluents.

2. Rules for the initial situation and goal.
   (a) relation $goal(S)$ is defined by the rule
   $$goal(S) \leftarrow holds(g, S).$$
   (b) The initial situation is defined by the collection of disjunctions of the form:
   $$h_1 \vee \ \ldots \ \vee h_k.$$
   where each h is a literal of the form $holds(f, 0)$ or $\neg holds(f, 0)$ where f is a fluent, and the awareness axiom
   $$holds(F, 0) \vee \neg holds(F, 0).$$

3. Rules for conformant planning:
   (a) Action generation. At any step, an action, if not prohibited in some belief set and the goal is not achieved in every belief set, may or may not occur.
   $$occurs(A, S) \vee \neg occurs(A, S) \leftarrow \neg M\ prohibited(A, S), \neg K\ goal(S).$$
   (b) At any step, only one action is allowed.
   $$\neg occurs(A_2, S) \leftarrow occurs(A_1, S), A_1 \neq A_2.$$
   (c) At any step, if an action occurs in a belief set, it occurs in every belief set, i.e., the same action occurs in every belief set.
   $$occurs(A, S) \leftarrow M\ occurs(A, S),\ S{<}m.$$
   (d) Add the constraint that at some step, the goal is achieved in every belief set.
   $$success \leftarrow K\ goal(S).$$
   $$\leftarrow\ not\ success.$$

To find a conformant plan, we need to find a sequence of actions such that no matter what the initial state is, the actions always achieve the goal. Intuitively, there is a one-one correspondence between the belief sets of a world view of the ES program above and the initial states. The step 3(a) applies the classical ASP method to generate actions, with a condition that action a is not prohibited by any belief set (i.e., $\neg M\ prohibited(a, S)$). The step 3(c) says that if an action occurs in one belief set of a world view, it should also occur in all the other belief sets of the world view. This rule naturally guarantees one sequence of actions is shared by all belief sets. The first rule of the step 3(d) says that if the goal is achieved in every belief set (i.e., from every possible initial state) at the same step, then we have $success$. These rules are natural extensions of the classical ASP rules for planning problems by straightforwardly adding the new requirement needed by conformant planning, demonstrating the elaboration tolerance capacity of our semantics. Our work is inspired by Kahl et al (2015)'s representation of this problem. The main difference is that their representation is not as elaboration tolerant (from planning problems to conformant planning problems) as ours. For example, in their work, the classical action generation rule is replaced by a new rule using M operator, and several involved rules are invented to assure the goal is achieved in the last step in every belief set.

The condition and correctness of our model of conformant planning problems is assured by the following result.

Proposition 2 *Given a conformant planning problem $\mathcal{P} = \langle D, \Sigma, g\rangle$ where D is consistent, deterministic and stable, A sequence $\alpha = \langle a_0, ..., a_j\rangle$, where $j < n$, of actions is a simple solution of $\mathcal{P}$ iff there is a world view W of $\tau_n(\mathcal{P})$ such that $occurs(a_k, k)$ $(k \in 0..j)$ belongs to its belief sets.*

Proof sketch. The proof is rather long and tedious and we only give a sketch here. Let Π be obtained from $\tau_n(\mathcal{P})$ by removing rules 3(d) in the definition of τ_n. We use Π^i $(i \in 0..n)$ to denote the rules of Π whose head literals have step i as their parameter. For each Π^i, we further divide it into Π^i_f: all rules defining $holds$, i.e., rules of the form step 1 (except those for executability conditions), 2(b) in the definition of τ_n; and Π^i_{gp}: all rules defining $goal$ and $prohibited$, i.e., rules for executability conditions and rules in step 2(a); Π^i_o: all rules defining $occurs$, i.e., rules of form 3(a)-(c).

We now prove the necessary condition $\Rightarrow$.

For each path $p = (\sigma_0, a_0, \ldots, a_j, \sigma_{j+1})$, where σ_0 is an initial state of $\mathcal{P}$, of $T(D)$, we define a function s which maps p to a set of literals as follows. For a state σ, we use $h(\sigma, i)$ to denote $\{holds(l, i) : l \in \sigma\}$. Let $GP(i)$ $(i \in 0..j+1)$ be the set of literals such that $h(\sigma_i, i) \cup GP(i)$ be the answer set of $h(\sigma_i, i) \cup \Pi^i_{gp}$. Let $notO(a_i, i) = \{\neg occurs(a, i) : a \neq a_i\}$. Intuitively, $GP(i)$ contains all literals with predicates of $goal$ and $prohibited$ derived from the state σ_i. Then $s(p)$ is defined as $(\cup_{i\in 0..j}(h(\sigma_i, i) \cup GP(i) \cup \{occurs(a_i, i)\} \cup notO(a_i, i))) \cup (\cup_{i\in j+1..n}(h(\sigma_{j+1}, i) \cup \{goal(i) : goal(j+1) \in GP(j+1)\} \cup \{prohibited(a, i) : prohibited(a, j+1) \in GP(j+1)\}$.

Let $W_1 = \{s(p) : p = (\sigma_0, a_0, \ldots, a_j, \sigma_{j+1})$, where σ_0 is an initial state of $\mathcal{P}$, is a path of $T(D)\}$.

We can show that W_1 is maximal and justified, and thus W_1 is a world view of Π. By the construction of W_1, $occurs(a_k, k)(k \in 0..j)$ belongs to each of its belief set.

We next prove the sufficient condition $\Leftarrow$.

Assume W is a world view of $\tau_n(\mathcal{P})$. We can show that the $occurs$ atoms of the belief sets of W coincide. Let $W_1 = \{A - \{success\} : A \in W\}$. By Lemma 1, W_1 is a world view of Π and there is some step s such that $W \models K\ goal(s)$. Let s be the smallest step such that $W \models K\ goal(s)$. We can show that there is an action a_i occurs at each step i for $i \in 0..s-1$ in each belief set of W_1. We can show that $a_0, \ldots, a_{s-1}$ is executable at any initial state σ_0. We can also show for any path $\sigma_0, a_0, \ldots, a_{s-1}, \sigma_s$, where σ_0 is an initial state, of $T(D)$, goal is achieved in σ_s. Hence, $a_0, \ldots, a_{s-1}$ is a solution of $\mathcal{P}$. Since s is the shortest step such that $W_1 \models K\ goal(s)$, no proper prefix of $a_0, \ldots, a_{s-1}$ is a solution of $\mathcal{P}$. Hence, $a_0, \ldots, a_{s-1}$ is a simple solution of $\mathcal{P}$. □

Lemma 1 used above, whose proof is in the supplement, shows that a program with those two rules in step 3(d) has a world view if and only if there exists s such that the goal is achieved at step s in every belief set of the world view.

Lemma 1 *Consider a program P, where $success$ does not occur, and rules $success \leftarrow K\ goal(S)$ and $\leftarrow not\ success$. Let $P' = P \cup \{success \leftarrow K\ goal(S). \leftarrow not\ success\}$. W_1 is a world view of P' iff $W_2 = \{S - \{success\} : S \in W_1\}$ is a world view of P and there is some s such that $W_1 \models K\ goal(s)$.*

Concluding Remarks

We develop a semantics for Epistemic Specifications using the idea of identifying each belief set with a program, and demonstrate the representation power of Epistemic Specifications on a class of conformant planning problems.

The semantics of Gelfond (2011) and Kahl et al. (2015) are based on new definitions of program reduct. Del Cerro et al. (2015) employ here-and-there logic to define epistemic equilibrium models and autoepistemic equilibrium models for an epistemic specification. Shen and Eiter (2016) propose a new semantics based on a so-called maximal guess of epistemic negation to minimize knowledge with respect to a world view. All these works eliminate circular justification to different extent. However, all of them allow $\{\{p\}\}$ to be a world view for program $\{p \leftarrow Mp\}$, while its only world view is $\{\{\}\}$ by our semantics.

It can be verified that our semantics for Epistemic Specifications coincides with ASP's for programs without modal operators. It is an interesting future work to study conditions when the different semantics coincide with each other. Our work will be continued in several directions: allowing nested and arbitrary modal formulas in a rule (del Cerro, Herzig, and Su 2015; Wang and Zhang 2005); studying the computational properties and identifying program classes with desirable complexity in practical applications; and developing a practical planner based on Epistemic Specifications under our semantics.

References

Baral, C., and Gelfond, M. 2000. *Reasoning Agents in Dynamic Domains*. Norwell, MA: Kluwer Academic Publishers. 257–279.

del Cerro, L.; Herzig, A.; and Su, E. 2015. Epistemic equilibrium logic. In *Proceedings of IJCAI-2015*, 2964–2970.

Erdem, E.; Lee, J.; and Lierler, Y. 2012. Theory and practice of answer set programming. *AAAI-2012 Tutorial (http://peace. eas. asu. edu/aaai12tutorial/asp-tutorial-aaai. pdf)*.

Gelfond, M., and Kahl, Y. 2014. *Knowledge Representation, Reasoning, and the Design of Intelligent Agents: The Answer-Set Programming Approach*. Cambridge University Press.

Gelfond, M. 1994. Logic programming with incomplete information. *Annals of Mathematics and Artificial Intelligence* 12(1-2):386–391.

Gelfond, M. 2011. New definition of epistemic specifications. In *Proceedings LPNMR-2011*, 260–265.

Kahl, P.; Watson, R.; Balai, E.; Gelfond, M.; and Zhang, Y. 2015. The language of epistemic specifications (refined) including a prototype solver. *Journal of Logic and Computation* exv065.

Kahl, P. 2014. *Refining the Semantics for Epistemic Logic Programs*. PhD Thesis.

Shen, Y.-D., and Eiter, T. 2016. Evaluating epistemic negation in answer set programming. *Artificial Intelligence* 237:115–135.

Truszczynski, M. 2011. Revisiting epistemic specifications. In *Logic Programming, Knowledge Representation and Nonmonotonic Reasoning*, 315–333.

Tu, P. H.; Son, T. C.; Gelfond, M.; and Morales, A. R. 2011. Approximation of action theories and its application to conformant planning. *Artificial Intelligence* 175(1):79–119.

Turner, H. 1997. Representing actions in logic programs and default theories: A situation calculus approach. *Journal of Logic Programming* 31(1–3):245–298.

Wang, K., and Zhang, Y. 2005. Nested epistemic logic programs. In *Proceedings of LPNMR-2005*, 279–290.