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 justifiedness 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 $\Pi_1$ 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 $\Pi_1$. 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 $\Pi_1$ 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 $\Pi_1$.

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 $\Pi_2$:

$$r_1 : p \leftarrow Mq, not\ q,$$

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

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 $\Pi_2$, without $M$ before them, can be understood

Copyright © 2017, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
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 $A_2$) $r_1': p_1 \leftarrow q_2, \neg q_1$. Note we don’t remove $Mq$ in this understanding because $q_2$ may in turn depend on beliefs in $A_2$. By replacing $Mq$ by $q_2$, circular justification can be taken care of by the resulting regular logic program(s). Since $q \not\in A_1$, we can remove $\neg q_1$ 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 only be understood, by $W$, as $p \not\in A_1$. However, since $p \in A_1$, $\neg p_1$ 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 $\Pi_{21}$ to denote the rule.

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

Since $\Pi_{21}$ refers to an atom in $\Pi_{22}$ which in turn refers to an atom in $\Pi_{21}$, we may not use $\Pi_{21}$ 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 $L$ 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 $ol$ to denote an extended subjective literal, where $\alpha$ 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 $W = \{A_1, \ldots, A_k\}$ 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 $\exists A_i \in W, l \in A_i$;
- $W \models Kl$ iff $W \not\models K l$;
- $W \models Kl$ iff $W \not\models M l$.

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

$$ l_1 \lor \cdots \lor l_k \leftarrow \neg l_{k+1}, \cdots, \neg l_m, \neg l_{m+1}, \cdots, \neg l_n, \quad (1) $$

where $l_1, \ldots, l_k$ and $l_{m+1}, \ldots, l_n$ are objective literals, and $l_1, \ldots, l_m$ 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 when $\neg l$ is satisfied in $A$, denoted as $A \models \neg l$, if $l \not\in A$. A disjunction of literals $l_1 \lor \cdots \lor l_k$ is satisfied in $A$, denoted as $A \models l_1 \lor \cdots \lor l_k$, 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 $\{l_{k+1}, \ldots, l_m\}$, $W \models l$, $A \models l'$ for all $A \in W$, and for each $l'' \in \{l_{m+1}, \ldots, l_n\}$, $A \models \neg l''$ for all $l'' \in W$, then $A \models l_1 \lor \cdots \lor l_k$ for all $A \in W$. Sometimes, we use "head($r$) \subseteq body($r$)" to represent $r$'s form (1). Also, if $r$'s body is empty, we simply represent (1) as "head($r$)". When "head($r$)" is empty, we call "body($r$)" a constraint.

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

Given a disjunctive program $\Pi$ and a belief set $A$, we say that a rule $r$ in $\Pi$ is satisfied in $A$ if $A \models \text{body($r$)}$ implies $A \models \text{head($r$)}$. $\Pi$ is satisfied in $A$, or $A$ is a model of $\Pi$, if each rule in $\Pi$ is satisfied in $A$. The reduct of $\Pi$ wrt $A$, denoted by $\Pi^A$, is the program obtained from $\Pi$ 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 $\Pi$ if $A$ is a minimal model of $\Pi^A$.

Disjunction Reduce

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 \( \Pi \) wrt a belief set \( A \), denoted by \( \Pi^{A,v} \), is a program resulting from \( \Pi \) by removing all literals not in \( A \) from the head of all the rules of \( \Pi \).

**Example 1** Consider program \( \Pi_3 = \{ p \lor q \} \). Given \( A_1 = \{ p \} \) and \( A_2 = \{ q \} \), then we have \( \Pi^{A_1,v}_3 = \{ q \} \) and \( \Pi^{A_2,v}_3 = \{ p \} \) 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 resulting 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 \( \Pi \) iff \( A \) satisfies all rules of \( \Pi \) and \( A \) is the answer set of the disjunction reduct of \( \Pi \) wrt \( A \).

**Justified Views**

In this section, we give a precise definition on justified views for ES programs. First, 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 \( \Pi \) and a collection \( W \) of belief sets \( \{A_1, ..., A_n\} \). Let \( EL_{11} \) be the set of all occurrences of extended subjective literals in \( \Pi \), and \( OL_{11} = \{ l_i | l_i \text{ is an objective literal occurring in } \Pi \text{ and } i \in 1..n \} \). A modal operator interpretation \( \rho \) for \( \Pi \) wrt \( W \) is a mapping \( \rho \) from \( EL_{11} \times 1..n \) to \( OL_{11} \), defined as follows:

\[
\forall i \rho (Kl, i) = l_i, \text{ if } W \models Kl; \tag{2}
\]
\[
\forall i \rho (Mi, i) = l_j, \text{ if } W \models Mi \text{ and } l \in A_j; \tag{3}
\]
\[
\forall i \rho (\neg Kl, i) = \neg l_j, \text{ if } W \models \neg Kl \text{ and } l \notin A_j; \tag{4}
\]
\[
\forall i \rho (\neg Mi, i) = \neg l_i, \text{ if } W \models \neg Mi. \tag{5}
\]

Let us take a closer look at Definition 2. Basically, mapping \( \rho \) 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 \( Mi \) 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 \( Mi \) being satisfied in \( W \) and assign \( \rho(Mi, i) = l_j \), as depicted in (3). The negative subjective literals such as \( \neg Kl \) and \( \neg Mi \) 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 \( ol \) 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 \( \Pi_4 \) as follows:

\[
p \leftarrow Mq, \text{ not } q; \tag{1}
\]
\[
q \leftarrow Mp, \text{ not } p; \tag{2}
\]
\[
\neg p, \text{ not } q. \tag{3}
\]

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

1. renaming each literal \( l \) not occurring in any subjective literal in \( \Pi \) by \( l_i \);
2. removing rules whose body contains \( ol \) such that \( W \not\models \neg ol \), and finally,
3. replacing every occurrence of extended subjective literal \( ol \) in the remaining program by \( \rho(ol, i) \).

Intuitively, by generating the modal reduct, we reduce the ES program \( \Pi \) to a disjunctive logic program \( \Pi^{W,A_i,\rho} \), in which all objective literals in \( \Pi \) 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 \( \rho \). The following example illustrates more details about this transformation.

**Example 3** [Example 2 continued] Still consider program \( \Pi_4 \) 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 \( \rho(Mp, i) = p_1 \) for \( i = 1,2 \) is a modal operator interpretation. Then we have the following two modal reducts: \( \Pi^{W,A_1,\rho} = \{ p_1 \leftarrow q_2, \text{ not } q_1, q_1 \leftarrow p_1, \text{ not } p_1, \text{ not } p_1, \text{ not } q_1 \} \), and \( \Pi^{W,A_2,\rho} = \{ p_2 \leftarrow q_2, \text{ not } q_2, p_1 \leftarrow q_2, \text{ not } q_2, \text{ not } p_2, \text{ not } q_2 \} \).

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

**Definition 4 (Justified view)** Consider an ES program \( \Pi \) and a collection \( W \) of belief sets \( \{A_1, ..., A_n\} \). Let \( B = \{ l_i | l \in A_i, i \in 1..n \} \). A full reduct of \( \Pi \) with respect to \( W \), \( A_i \) and a modal operator interpretation \( \rho \), denoted by \( \Pi^{W,A_i,\rho,\not\not\not,\not\not\not} \), is the program obtained by applying modal reduct based on \( W \), \( A_i \) and \( \rho \), Gelfond-Lifschitz
reduct and disjunction reduct with respect to $B$ in sequence to $\Pi$: $(\Pi_{1}^{W,A,\rho})^{B}$. $W$ is a justified view of $\Pi$ if there exists a modal operator interpretation $\rho$ such that $B$ is the answer set of the program $\bigcup_{i=1}^{n} \Pi_{i}^{W,A,\rho}$, not $\forall$.

Example 4 [Example 3 continued] Let us consider program $\Pi_{4}$ in Example 2 once again. As in Example 3, let $W = \{A_{1}, A_{2}\}$, where $A_{1} = \{p\}$, $A_{2} = \{q\}$, and $\rho(Mg,i) = q_{2}$ and $\rho(Mp,i) = p_{1}$, otherwise. Then we have $\Pi_{4}^{W,A,\rho}$ and $\Pi_{4}^{W,A,\rho}$ not $\forall$ 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 $(\Pi_{4}^{W,A,\rho})^{B} = \{p_{1} \leftarrow q_{2}\}$, and $(\Pi_{4}^{W,A,\rho})^{B} = \{q_{2} \leftarrow p_{1}\}$. Since the heads of the rules do not have disjunctions, the programs above keep the same after disjunction reduct. Therefore, $\Pi_{4}^{W,A,\rho}$, not $\forall$ $= \{p_{1} \leftarrow q_{2}\}$, denoted by $\Pi_{41}$, and $\Pi_{4}^{W,A,\rho}$, not $\forall$ $= \{q_{2} \leftarrow p_{1}\}$, denoted by $\Pi_{42}$. 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 $W' \neq Mp$ and $W' \neq Mq$, no modal interpretation is needed. So the only full reduct is: $\{\}$. Thus, this example shows that $\Pi_{4}$ does not have any justified view.

Without getting into details, we can also show that program $\Pi_{2}$, which is illustrated in the introduction section and the same as $\Pi_{4}$ except not including the constraint “$\forall$ - not $p$, not $q$”, has a unique justified view $\{\}$, according to our previous definitions.

Example 5 Consider the program $\Pi_{5}$:

\[
\begin{align*}
p \lor q, \\
qu \lor s, \\
an \leftarrow \neg Mp, \\
b \leftarrow \neg Ms, \\
c \leftarrow \neg a, \\
d \leftarrow \neg b.
\end{align*}
\]

Let $W_{1} = \{p, s, c, d\}, \{q, c, d\}$ and $W_{2} = \{q, a, b\}$. Since $W_{1} \neq Mp$ and $W_{1} \neq Ms$, there is “no” modal operator interpretation for extended subjective literals in $\Pi_{1}$ under $W_{1}$; while $\rho(\neg Mp, 1) = \neg p_{1}$ and $\rho(\neg Ms, 1) = \neg s_{1}$ 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 $\Pi_{51}$ and $\Pi_{52}$ 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 $\Pi_{51} \cup \Pi_{52} = B = \{p_{1}, s_{1}, c_{1}, d_{1}, q_{2}, c_{2}, d_{2}\}$. $W_{1}$ is a justified view of $\Pi_{5}$. 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 $\Pi_{5}$.

In fact, we can further show that $W_{1}$ and $W_{2}$ are the only two justified views of program $\Pi_{5}$.

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 $\Pi_{6}$ consisting of a single rule:

\[
p \lor q.
\]

It is not hard to see that $\Pi_{6}$ has three justified views: $\{\{p\}\}$, $\{\{q\}\}$ and $\{\{p\}, \{q\}\}$. Obviously only the last one should be a rational model for $\Pi_{6}$.

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 provide a semantic foundation for ES programs.

Definition 5 (Maximal view) Let $\Pi$ be an ES program and $W = \{A_{1}, \cdots, A_{n}\}$, where $A_{1}, \cdots, A_{n}$ are belief sets. A disjunctive program $\Pi^{W}$ is called the general modal reduct of $\Pi$ with respect to $W$, denoted by $\Pi^{W}$, if it is obtained from $\Pi$ 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 $\Pi$;
2. for every $Ml$ occurring in $r$, removing it from $r$’s body if $W \models Ml$, otherwise removing $r$ from $\Pi$;
3. for every $\neg Kl$ occurring in $r$, removing it from $r$’s body if $W \models \neg Kl$, otherwise removing $r$ from $\Pi$;
4. for every $\neg Ml$ occurring in $r$, replacing it by not $l$ if $W \models \neg Ml$, otherwise removing $r$ from $\Pi$.

We call $W$ a maximal view if $W$ is the collection of all answer sets of $\Pi^{W}$.

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 $\alpha l$ 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 equivalent to replace it by $T$, 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 $\Pi_{6}$. 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 \( \Pi \) be an ES program and \( W \) a collection of belief sets. \( W \) is a world view of \( \Pi \) if \( W \) is a justified and maximal view of \( \Pi \).

**Example 6** Consider Program \( \Pi_7 \):

\[
\begin{align*}
p \vee q, \\
q \vee r, \\
p &\leftarrow Mp, \\
s &\leftarrow p, q, \\
s &\leftarrow Ms, \\
\leftarrow p, \text{not} \ s.
\end{align*}
\]

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

From Definitions 2 and 3, we can also obtain the two modal reducts \( \{ p, q, s \} \) and \( \{ p, r, s \} \) as follows:

\[
\begin{align*}
\{ p_1 \vee q_1, q_1, p_1 &\leftarrow p_2, s_1 \leftarrow p_1, q_1, s_1 \leftarrow p_1, \text{not} \ s_1 \}, \\
\{ p_2, r_2, p_2 &\leftarrow p_2, s_2 \leftarrow p_2, q_2, s_2 \leftarrow s_1 \leftarrow p_2, \text{not} \ s_2 \}.
\end{align*}
\]

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 \( \Pi_7 \).

**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 \( \Pi(D, \sigma, \{ \alpha \}) \), where \( D \) is a system description, \( \sigma \) a state and \( \{ \alpha \} \) a set of actions, to denote the ASP program \( \Pi(D) \cup \{ \text{holds}(l, 0) : l \in \sigma \} \cup \{ \text{occurs}(a_i, 0) : a_i \in \alpha \} \) where \( \Pi(D) \) is an ASP program obtained from \( D \) as in (Gelfond and Kahl 2014).

Actions of \( \alpha \) are said to be prohibited in a state \( \sigma \) 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 \( \sigma \); otherwise, actions in \( \alpha \) are said to be executable in \( \sigma \). A sequence of actions \( a_0, \ldots, a_{n-1} \) is executable in a state \( \sigma \) if the sequence is empty or \( a_0 \) is executable in \( \sigma \) and the sequence \( a_1, \ldots, a_{n-1} \) is executable in \( \sigma' \) for every \( (\sigma, a_0, \sigma') \in T(D) \).

A system description \( D \) is consistent if for any state \( \sigma_1 \) and an action \( a \) executable in \( \sigma_1 \), there exists at least one state \( \sigma_2 \) such that \( (\sigma_1, a, \sigma_2) \in T(D) \). A system description is deterministic if for any state \( \sigma_1 \) and action \( a \), there is at most one state \( \sigma_2 \) such that \( (\sigma_1, a, \sigma_2) \) is a transition defined by \( T(D) \). A system description \( D \) is stable if for any state \( \sigma_1 \), \( \Pi(D, \sigma, \{ \} \) has a unique answer set \( A \) and \( \sigma = \{ l : \text{holds}(l, 1) \in A \} \). 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 \( (D, \Sigma, g) \) which consists of a system description \( D \) of an action language \( \mathcal{AL} \), a collection of the possible initial states \( \Sigma \), and a goal \( g \) where \( g \) is a set of fluent literals. A sequence \( \alpha = \{ a_0, \ldots, a_{n-1} \} \) of actions is called a solution to \( P \) if \( \alpha \) is executable in every state of \( \Sigma \) 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 \( \sigma_n \), i.e., \( g \subseteq \sigma_n \). A solution \( \alpha \) of a conformant planning problem \( P \) is simple if no proper prefix of \( \alpha \) is a solution of \( P \).

Given a conformant planning problem \( P = (D, \Sigma, g) \), 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 executibility condition which are translated as follows: for each executibility condition

   \[
   \text{impossible} \ a \text{ if } l_1, \ldots, l_n,
   \]

   it is translated into

   \[
   \text{prohibited}(a, S) \leftarrow \text{holds}(l_1, S), \ldots, \text{holds}(l_n, S)
   \]

   where \( \text{prohibited} \) is a new predicate, \( \text{prohibited}(a, S) \) denotes action \( a \) is prohibited at step \( S \) and \( \text{holds}(l, S) \) 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 \( \text{goal}(S) \) is defined by the rule

   \[
   \text{goal}(S) \leftarrow \text{holds}(g, S).
   \]

   (b) The initial situation is defined by the collection of disjunctions of the form:

   \[
   h_1 \lor \ldots \lor h_k,
   \]

   where each \( h \) is a literal of the form \( \text{holds}(f, 0) \) or \( \lnot \text{holds}(f, 0) \) where \( f \) is a fluent, and the awareness axiom

   \[
   \text{holds}(F, 0) \lor \lnot \text{holds}(F, 0).
   \]

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.

   \[
   \text{occurs}(A, S) \lor \lnot \text{occurs}(A, S) \leftarrow \lnot M \text{ prohibited}(A, S), \lnot K \text{ goal}(S).
   \]

   (b) At any step, only one action is allowed.

   \[
   \lnot \text{occurs}(A_2, S) \leftarrow \text{occurs}(A_1, S), A_1 \neq A_2.
   \]

   (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.

   \[
   \text{occurs}(A, S) \leftarrow M \text{ occurs}(A, S), S < m.
   \]

   (d) Add the constraint that at some step, the goal is achieved in every belief set.

   \[
   \text{success} \leftarrow K \text{ goal}(S).
   \]

   \[
   \lnot \text{not success}.
   \]
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 \( \mathcal{P} = (D, \Sigma, g) \) 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 \( \text{occurs}(a_k, k) (k \in 0..j) \) belongs to its belief sets.

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

For each path \( p = (\sigma_0, a_0, ..., a_j, \sigma_{j+1}) \), where \( \sigma_0 \) 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 \( \sigma \), we use \( h(\sigma, i) \) to denote \{\( \text{holds}(l, i) : l \in \sigma \)\}. Let \( \text{GP}(i) (i \in 0..j + 1) \) be the set of literals such that \( h(\sigma, i) \cup \text{GP}(i) \) be the answer set of \( h(\sigma, i) \cup \Pi^i_{\Pi^i} \). Let \( \text{notO}(a, i) = \{\neg \text{occurs}(a, i) : a \neq a_i \} \). Intuitively, \( \text{GP}(i) \) contains all literals with predicates of \( \text{goal} \) and \( \text{prohibited} \) derived from the state \( \sigma_i \). Then \( s(p) \) is defined as \( \langle \cup_{i \in 0..j} (h(\sigma, i) \cup \text{GP}(i) \cup \text{occurs}(a, i) \cup \neg \text{notO}(a, i)) \rangle \cup \langle \cup_{i \in j+1..n} (h(\sigma, i) \cup \{ \text{goal}(i) : \text{goal}(i + 1) \in \text{GP}(i + 1) \}) \cup \{ \text{prohibited}(a, i) : \text{prohibited}(a, i + 1) \in \text{GP}(i + 1) \}) \rangle \).

Let \( \mathcal{W}_1 = \{ s(p) : p = (\sigma_0, a_0, ..., a_j, \sigma_{j+1}), \text{where } \sigma_0 \text{ is an initial state of } \mathcal{P}, \text{is a path of } T(D) \} \).

We can show that \( \mathcal{W}_1 \) is maximal and justified, and thus \( \mathcal{W}_1 \) is a world view of \( \Pi \). By the construction of \( \mathcal{W}_1 \), \( \text{occurs}(a_k, k) (k \in 0..j) \) belongs to each of its belief set.

We next prove the sufficient condition \( \Leftarrow \).

Assume \( \mathcal{W} \) is a world view of \( \tau_n(\mathcal{P}) \). We can show that the \( \text{occurs} \) atoms of the belief sets of \( W \) coincide. Let \( \mathcal{W}_1 = \{ A - \{ \text{success} \} : A \in \mathcal{W} \} \). By Lemma 1, \( \mathcal{W}_1 \) is a world view of \( \Pi \) and there is some step \( s \) such that \( W \models K \text{goal}(s) \). Let \( s \) be the smallest step such that \( W \models K \text{goal}(s) \). We can show that there is an action \( a \) occurs at each step \( i \) for \( i \in 0..s - 1 \) in each belief set of \( \mathcal{W}_1 \). We can show that \( a_0, ..., a_{s-1} \) is executable at any initial state \( \sigma_0 \). We can also show for any path \( \sigma_0, a_0, ..., a_{s-1}, \sigma_s \), where \( \sigma_0 \) is an initial state, \( T(D) \), goal is achieved in \( \sigma_s \). Hence, \( a_0, ..., a_{s-1} \) is a solution of \( \mathcal{P} \). Since \( s \) is the shortest step such that \( \mathcal{W}_1 \models K \text{goal}(s) \), no proper prefix of \( a_0, ..., a_{s-1} \) is a solution of \( \mathcal{P} \). Hence, \( a_0, ..., a_{s-1} \) 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 \( \mathcal{P} \), where \text{success} does not occur, and rules success \( \Leftarrow K \text{goal}(S) \) and \text{not success}, \text{success} \( \Leftarrow K \text{goal}(S) \), \( \Leftarrow \text{not success} \). \( \mathcal{W}_1 \) is a world view of \( \mathcal{P} \) iff \( \mathcal{W}_2 = \{ S - \{ \text{success} \} : S \in \mathcal{W}_1 \} \) is a world view of \( \mathcal{P} \) and there is some \( s \) such that \( \mathcal{W}_1 \models K \text{goal}(s) \).

**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 \{ \{ \} \} to be a world view for program \{ \{ \} \}, 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