Artificial Intelligence 237 (2016) 115–135

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Evaluating epistemic negation in answer set programming

Yi-Dong Shen [a,*], Thomas Eiter [b]

[a] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
[b] Institut für Informationssysteme, Technische Universität Wien, Favoritenstrasse 9-11, A-1040 Vienna, Austria

ARTICLE INFO

Article history:
Received 15 August 2015
Received in revised form 31 December 2015
Accepted 12 April 2016
Available online 22 April 2016

Keywords:
Answer set programming
Epistemic negation
Semantics

ABSTRACT

Epistemic negation **not** along with default negation ¬ plays a key role in knowledge representation and nonmonotonic reasoning. However, the existing epistemic approaches such as those by Gelfond [13,15,14], Truszczynski [33] and Kahl et al. [18] behave not satisfactorily in that they suffer from the problems of unintended world views due to recursion through the epistemic modal operator **K** or **M** (**K***F* and **M***F* are shorthands for ¬**not** *F* and **not**¬*F*, respectively). In this paper we present a new approach to handling epistemic negation which is free of unintended world views and thus offers a solution to the long-standing problem of epistemic specifications which were introduced by Gelfond [13] over two decades ago. We consider general logic programs consisting of rules of the form $H \leftarrow B$, where H and B are arbitrary first-order formulas possibly containing epistemic negation, and define a general epistemic answer set semantics for general logic programs by introducing a novel program transformation and a new definition of world views in which we apply epistemic negation to minimize the knowledge in world views. The general epistemic semantics is applicable to extend any existing answer set semantics, such as those defined in [26,27,32,1,8,12,29], with epistemic negation. For illustration, we extend FLP answer set semantics of Faber et al. [8] for general logic programs with epistemic negation, leading to epistemic FLP semantics. We also extend the more restrictive well-justified FLP semantics of Shen et al. [29], which is free of circularity for default negation, to an epistemic well-justified semantics. We consider the computational complexity of epistemic FLP semantics and show that for a propositional program Π with epistemic negation, deciding whether Π has epistemic FLP answer sets is Σ_3^p-complete and deciding whether a propositional formula F is true in Π under epistemic FLP semantics is Σ_4^p-complete in general, but has lower complexity for logic programs that match normal epistemic specifications, where the complexity of world view existence and query evaluation drops by one level in the polynomial hierarchy.

1. Introduction

Answer set programming (ASP) is a major logic programming paradigm rooted in knowledge representation and reasoning (KR) for modeling and solving knowledge-intensive search and optimization problems such as product configuration and planning [2]. In ASP, the semantics of a logic program is given by a set of intended models, called stable models or answer sets [16,17]. Such answer sets can be defined in different ways; Lifschitz [21] listed thirteen of them in the literature. These

* Corresponding author.
E-mail addresses: ydshen@ios.ac.cn (Y.-D. Shen), eiter@kr.tuwien.ac.at (T. Eiter).

http://dx.doi.org/10.1016/j.artint.2016.04.004

semantics agree for normal logic programs, but show discrepancies for more general logic programs such as logic programs with aggregates [30,8], with external sources such as description logic programs (dl-programs) [7], and with propositional or first-order formulas [26,27,32,1,12]. Most recently Shen et al. [29] introduced a new one called the *well-justified FLP answer set semantics*, which is fundamentally distinct from other existing answer set semantics in that every answer set of a general logic program is justified by having a level mapping and thus is free of circular justifications. This semantics has been implemented over the well-known ASP reasoner DLVHEX.[1]

Negation is a key mechanism in ASP for reasoning with incomplete knowledge. There are two major types of negation, *default negation* and *epistemic negation*. A third, called *strong negation*, also appears in the literature; when default negation is available, strong negation is easily compiled away using new predicate symbols [17] and thus it can be omitted. By abuse of notation, in this paper we use $\neg$, **not** and $\sim$ to denote the three negation operators, respectively.[2] For a formula F, the default negation $\neg F$ of formula F expresses that there is no *justification* for adopting F in an answer set and thus F can be assumed false by default in the answer set; in contrast, the epistemic negation **not** F of F expresses that there is no evidence *proving* that F is true, i.e., F is false in some answer set. *Justification* in ASP is a concept defined over every individual answer set, while *provability* is a meta-level concept defined over a collection of answer sets, called a *world view*. This means the two types of negation are orthogonal operations, where default negation works *locally* on each individual answer set, and epistemic negation works *globally* at a meta level on each world view.

With both default and epistemic negation, ASP is enabled to reason with different incomplete knowledge. For example, we can use the rule

$$innocent(X) \leftarrow \mathbf{not}\, guilty(X)$$

to concisely express the *presumption of innocence*, which states that one is presumed innocent if there is no evidence proving s/he is guilty. We can also use rules of the form

$$\neg p(X) \leftarrow \mathbf{not}\, p(X)$$

to explicitly state Reiter's *closed-world assumption* (CWA) [28], i.e., if there is no evidence proving $p(X)$ is true we jump to the conclusion that $p(X)$ is false.

However, observe that most of the existing answer set semantics, such as those defined in [17,26,27,32,1,8,12,29], only support default negation and they do not allow for epistemic negation.

Epistemic negation and specifications. In fact, the need for epistemic negation was long recognized in ASP by Gelfond in the early 1990s [13,15] and recently revisited in [14,33,19,18,4]. In particular, Gelfond [13] showed that formalization of CWA using default and strong negations with rules of the form

$$\sim p(X) \leftarrow \neg p(X)$$

as presented in [17], is problematic.[3] He then proposed to address the problem using two epistemic modal operators **K** and **M**. Informally, for a formula F, $\mathbf{K}F$ expresses that F is true in every answer set, and $\mathbf{M}F$ expresses that F is true in some answer set. Note that $\mathbf{M}F$ can be viewed as shorthand for $\neg\mathbf{K}\neg F$.[4]

In the sequel, by an *object literal* we refer to an atom A or its strong negation $\sim A$; a *default negated literal* is of the form $\neg L$, and a *modal literal* is of the form $\mathbf{K}L$, $\neg\mathbf{K}L$, $\mathbf{M}L$ or $\neg\mathbf{M}L$, where L is an object literal.

Gelfond [13] considered disjunctive logic programs with modal literals, called *epistemic specifications*, which consist of rules of the form

$$L_1 \vee \cdots \vee L_m \leftarrow G_1 \wedge \cdots \wedge G_n \tag{1}$$

where each L is an object literal and each G is an object literal, a default negated literal, or a modal literal. A *normal epistemic specification* consists of rules of the above form with $m = 1$. Given a collection $\mathcal{A}$ of interpretations as an *assumption*, a logic program Π is transformed into a *modal reduct* $\Pi^{\mathcal{A}}$ w.r.t. the assumption $\mathcal{A}$ by first removing all rules with a modal literal G that is not true in $\mathcal{A}$, then removing the remaining modal literals. The assumption $\mathcal{A}$ is defined to be a *world view* of Π if it coincides with the collection of answer sets of $\Pi^{\mathcal{A}}$ under the semantics defined in [17].

The problem with recursion through K. More recently, Gelfond [14] addressed the problem that applying the above approach to handle modal literals may produce unintuitive world views due to recursion through **K**. For example, consider a logic program $\Pi = \{p \leftarrow \mathbf{K}p\}$. The rule expresses that for any collection $\mathcal{A}$ of answer sets of Π and any $I \in \mathcal{A}$, if p is true in all answer sets in $\mathcal{A}$, then p is true in I. This amounts to saying that if p is true in all answer sets, then p is always true (in particular in all answer sets). Obviously, this rule is not informative and does not contribute to constructively building any answer set; thus it can be eliminated from Π, leading to $\Pi = \emptyset$. As a result, Π is expected to have a unique answer

[1] www.kr.tuwien.ac.at/research/systems/dlvhex.

[2] In many texts, *not* and $\neg$ are used to denote the default and strong negation operators, respectively.

[3] We will further discuss this issue in Remark 3 following Example 4.

[4] Note that $\neg\mathbf{K}F$ and $\sim\mathbf{K}F$ are semantically equivalent. In [13], $\mathbf{M}F$ is shorthand for $\sim\mathbf{K}\sim F$, while in [14], it is shorthand for $\sim\mathbf{K}\neg F$, which is semantically equivalent to $\neg\mathbf{K}\neg F$.

set $\emptyset$. However, $\{p\}$ would be an answer set of Π when applying the approach of Gelfond [13]. To illustrate, consider an assumption $\mathcal{A} = \{\{p\}\}$, i.e., p is assumed to be true in all interpretations in $\mathcal{A}$. Then, $\mathbf{K}p$ is true in $\mathcal{A}$ and we obtain the modal reduct $\Pi^{\mathcal{A}} = \{p\}$. This reduct has a unique answer set $\{p\}$, which coincides with the assumption $\mathcal{A}$. Thus $\mathcal{A}$ is a world view of Π under Gelfond [13]. Observe that this world view has an epistemic circular justification that can be expressed as

$$\exists_{I\in\mathcal{A}}\ p \in I \ \Leftarrow\ \mathbf{K}p \ \Leftarrow\ \forall_{I\in\mathcal{A}}\ p \in I \tag{2}$$

where the arrow $\Leftarrow$ stands for "is due to." That is, p being true in an interpretation $I = \{p\}$ of the world view $\mathcal{A}$ is due to $\mathbf{K}p$ being treated true in the program transformation for the modal reduct $\Pi^{\mathcal{A}}$ (via the rule $p \leftarrow \mathbf{K}p$), which in turn is due to p being assumed to be true in all interpretations of $\mathcal{A}$.

In general, a world view $\mathcal{A}$ is said to have an *epistemic circular justification* if some object literal L being true in some interpretation $I \in \mathcal{A}$ is due to $\mathbf{K}L$ (or its equivalent modal literals expressing that L is true in every interpretation $J \in \mathcal{A}$) being treated true in the program transformation for the modal reduct of Π w.r.t. $\mathcal{A}$. This means that L being true in some interpretation of $\mathcal{A}$ is due to L being assumed to be true in all interpretations of $\mathcal{A}$.

To remedy the epistemic circular justification problem with recursion through **K**, Gelfond [14] revised the program transformation such that a modal reduct $\Pi^{\mathcal{A}}$ is obtained from Π by first removing all rules of form (1) with a modal literal G that is not true in $\mathcal{A}$, then removing all modal literals $\neg\mathbf{K}L$ and $\mathbf{M}L$, and finally replacing all modal literals $\mathbf{K}L$ by L and $\neg\mathbf{M}L$ by $\neg L$.

It is easy to check that the logic program $\Pi = \{p \leftarrow \mathbf{K}p\}$ has a unique world view $\{\emptyset\}$ when applying the revised program transformation. Unfortunately, the epistemic circular justification problem persists in other logic programs, such as $\Pi = \{q \leftarrow \neg\mathbf{K}p,\ p \leftarrow \neg q\}$. Consider an assumption $\mathcal{A} = \{\{p\}\}$. Since $\mathbf{K}p$ is true in $\mathcal{A}$, the modal literal $\neg\mathbf{K}p$ is not true in $\mathcal{A}$ and thus the first rule is removed, yielding the modal reduct $\Pi^{\mathcal{A}} = \{p \leftarrow \neg q\}$. This reduct has a unique answer set $\{p\}$, which coincides with $\mathcal{A}$, hence $\mathcal{A}$ is a world view of Π. Note that this world view has also an epistemic circular justification

$$\exists_{I\in\mathcal{A}}\ p \in I \ \Leftarrow\ \neg q \ \Leftarrow\ \neg\mathbf{K}p \ \Leftarrow\ \forall_{I\in\mathcal{A}}\ p \in I,$$

i.e., p being true in an interpretation $I = \{p\}$ of the world view $\mathcal{A}$ is (via the rule $p \leftarrow \neg q$) due to q being false in I, which in turn (via the rule $q \leftarrow \neg\mathbf{K}p$) is due to $\mathbf{K}p$ being treated true and thus $\neg\mathbf{K}p$ treated false in the program transformation, which is due to p being assumed to be true in all interpretations of $\mathcal{A}$.

The problem with recursion through M. In addition to the problem of unintended world views due to recursion through **K**, the approaches of Gelfond [14,13] also suffer from the problem of unintended world views due to recursion through **M**. Consider the logic program $\Pi = \{p \leftarrow \mathbf{M}p\}$, which expresses that for any world view $\mathcal{A}$ and any $I \in \mathcal{A}$, if p is true in some answer set in $\mathcal{A}$, then p is true in I. This amounts to saying that if p is true in some answer set, then p is always true, in particular in every answer set. Under the approaches of Gelfond [14,13] this program has two world views, $\{\{p\}\}$ and $\{\emptyset\}$. Naturally, the question is whether both are intuitive; ideally, we have only one world view. If, for example, p expresses "something goes wrong," then the program could be viewed as a paraphrase of *Murphy's law*: "if something can go wrong, it will go wrong," and accordingly, the intuitive world view would be $\{\{p\}\}$.

Recent advance. Recent work of Kahl et al. [19,18] and del Cerro et al. [4] suggests that indeed $\{\{p\}\}$ should be the only world view of the program $\Pi = \{p \leftarrow \mathbf{M}p\}$. The supporting intuition is twofold. First, there seems no justification that this program has two world views, one derived from treating the modal literal $\mathbf{M}p$ to be true and the other from treating it false. Second, Kahl et al. [18] observed that there is a *preference order* over $\mathbf{K}p$, p and $\mathbf{M}p$ in terms of the degree of conviction in establishing them w.r.t. a world view $\mathcal{A}$: in order to establish $\mathbf{K}p$, it must be demonstrated that p belongs to *all* answer sets in $\mathcal{A}$; to establish p, it must be demonstrated that p belongs to a *particular* answer set in $\mathcal{A}$; and to establish $\mathbf{M}p$, it is sufficient to demonstrate that p belongs to *some* answer set in $\mathcal{A}$. This means $\mathbf{K}p$ is of the highest conviction and thus is harder to establish than p that is harder than $\mathbf{M}p$ that is of the lowest conviction. Then, intuitively stronger literals are expected to be established from or supported by weaker ones, i.e., $\mathbf{K}p$ can be supported by p that can be supported by $\mathbf{M}p$ without incurring epistemic circular justifications. This suggests that the case that some object literal L being true in a *particular* answer set $I \in \mathcal{A}$ is due to $\mathbf{M}L$ (or its equivalent modal literals expressing that L is true in *some* answer set $J \in \mathcal{A}$) being treated true in the program transformation for the modal reduct of Π w.r.t. $\mathcal{A}$ does not make an epistemic circular justification. Kahl et al. [18] then used this intuition to justify $\{\{p\}\}$ to be a world view of the logic program $\{p \leftarrow \mathbf{M}p\}$, where p is supported by $\mathbf{M}p$.

In fact, Kahl et al. [19,18] extensively studied the problems of unintended world views due to recursion through **K** and **M** with the approaches of Gelfond [14,13] and proposed a new program transformation by appealing to *nested expressions* defined by Lifschitz et al. [22]. Let Π be a logic program with rules of form (1) and $\mathcal{A}$ a collection of interpretations as an assumption. The *Kahl modal reduct* $\Pi^{\mathcal{A}}$ of Π w.r.t. $\mathcal{A}$ is a nested logic program (i.e., a logic program with nested expressions), which is obtained from Π by replacing all modal literals or deleting rules according to Table 1. The assumption $\mathcal{A}$ is defined to be a *world view* of Π if it coincides with the collection of answer sets of the nested logic program $\Pi^{\mathcal{A}}$ under the semantics of Lifschitz et al. [22].

Kahl [19] collected sixty-two interesting logic programs with modal literals and illustrated the approach with these programs. As a typical example, given an assumption $\mathcal{A} = \{\emptyset\}$, the Kahl modal reduct of the logic program $\Pi = \{p \leftarrow \mathbf{M}p\}$

Table 1
The program transformation defined by Kahl et al. [19,18], where G is a modal literal in the body of a rule. Note that $\neg\neg L$ is a nested expression defined by Lifschitz et al. [22], which is not equivalent to L as in classical logic.

G	G is true in $\mathcal{A}$	Otherwise
$\mathbf{K}L$	replace it with L	delete the rule
$\neg\mathbf{K}L$	replace it with $\top$	replace it with $\neg L$
$\mathbf{M}L$	replace it with $\top$	replace it with $\neg\neg L$
$\neg\mathbf{M}L$	replace it with $\neg L$	delete the rule

w.r.t. $\mathcal{A}$ is $\Pi^{\mathcal{A}} = \{p \leftarrow \neg\neg p\}$, which has two answer sets $\emptyset$ and $\{p\}$ under the semantics of Lifschitz et al. [22]. Thus $\{\emptyset\}$ is not a world view. Consequently this program has a unique world view $\{\{p\}\}$ under the approach of Kahl et al. [19,18].

However, our careful study reveals three critical shortcomings of this approach:

(1) The definition of the Kahl program transformation/modal reduct looks a bit ad hoc, and the variety of replacements for modal literals (see Table 1) lacks a deeper discussion or justification.[5]
(2) It is undesired to transform a logic program into a reduct containing nested expressions. As shown in Shen et al. [29], the existing semantics for nested expressions, such as those defined in [22,11,12], suffer from circular justifications. For example, for the logic program $\Pi = \{p \leftarrow \neg\neg p\}$, $I = \{p\}$ is an answer set under these semantics. Observe that this answer set has a circular justification via the self-supporting loop

$$p \in I \Leftarrow \neg\neg p \Leftarrow p \in I \tag{3}$$

i.e., p being true in I is due to I satisfying $\neg\neg p$ (via the rule $p \leftarrow \neg\neg p$), which in turn is due to p being true in I.[6] For a logic program with rules of form (1), it is desirable to transform it to a regular disjunctive logic program so that the standard answer set semantics by Gelfond and Lifschitz [16,17] can be applied.
(3) We observe that applying the Kahl program transformation to some logic programs with recursion through **M** may also produce unintended world views, as illustrated in the following example.

Example 1. Consider the following logic program, which is borrowed from Example 29 in Appendix D of Kahl [19]:

$$\Pi: \quad p \leftarrow \mathbf{M}q \wedge \neg q \qquad r_1$$
$$q \leftarrow \mathbf{M}p \wedge \neg p \qquad r_2$$

This program is very similar to $\{p \leftarrow \mathbf{M}p\}$ in the way that the modal operator **M** is used to recursively support the rule heads. Under the approaches of Gelfond [14,13] this program has two world views, viz. $\mathcal{A}_1 = \{\{p\}, \{q\}\}$ and $\mathcal{A}_2 = \{\emptyset\}$, where $\mathcal{A}_1$ is derived from treating $\mathbf{M}p$ and $\mathbf{M}q$ to be true and $\mathcal{A}_2$ derived from treating them false.

Following the same intuition of Kahl et al. [19,18] for $\{p \leftarrow \mathbf{M}p\}$ as described above, $\mathcal{A}_1$ is expected to be the only world view of this program. However, applying the Kahl program transformation to this program will produce the two world views $\mathcal{A}_1$ and $\mathcal{A}_2$.

Our contributions. The goal of this paper is to address the above problems of unintended world views and provide a satisfactory solution to epistemic negation as well as epistemic specifications of Gelfond [13]. Our main contributions are summarized as follows:

(1) We use modal operator **not** to directly express epistemic negation and define general logic programs consisting of rules of the form $H \leftarrow B$, where H and B are arbitrary first-order formulas possibly containing epistemic negation. Modal formulas $\mathbf{K}F$ and $\mathbf{M}F$ are viewed as shorthands for $\neg\,\mathbf{not}\,F$ and $\mathbf{not}\,\neg F$, respectively, and thus epistemic specifications of Gelfond [13] are a special class of general logic programs.
(2) We propose to apply epistemic negation to minimize the knowledge in world views of a general logic program Π, i.e., we apply epistemic negation to arbitrary closed first-order formulas F w.r.t. a world view and assume $\mathbf{not}\,F$ in Π to be true in the world view whenever possible; we refer to this idea as *knowledge minimization with epistemic negation*. It is analogous to applying default negation to minimize the knowledge in answer sets, i.e., one applies default negation to arbitrary ground atoms A w.r.t. an answer set and assumes $\neg A$ to be true in the answer set whenever possible (CWA or minimal models); this is referred to as *knowledge minimization with default negation*. To this end, we introduce a novel and very simple program transformation based on epistemic negation and present a new definition of world views.

[5] In fact, no existing approaches to epistemic specifications, such as those of Gelfond [13,14], Truszczynski [33], and Kahl et al. [19,18], have ever provided a deeper discussion or justification for the replacements of modal literals in their program transformations.

[6] Note the difference between a circular justification of form (3), which says that p being true in an answer set I is due to p being assumed to be true in I, and an epistemic circular justification of form (2), which says that p being true in some answer set I of a world view $\mathcal{A}$ (i.e., $\exists_{I\in\mathcal{A}}\ p \in I$) is due to p being assumed to be true in every answer set I in $\mathcal{A}$ (i.e., $\forall_{I\in\mathcal{A}}\ p \in I$).

Given a subset Φ of the epistemic negations **not** F in Π, called a *guess*, we transform Π into an *epistemic reduct* based on Φ, denoted Π^{Φ}, by replacing every **not** F with $\top$ if it is in Φ, and with $\neg F$, otherwise. Let $\mathcal{A}$ be the set of all answer sets of Π^{Φ}. Then we call $\mathcal{A}$ a *candidate world view* w.r.t. Φ if it agrees with Φ in the sense that every **not** F in Π is true in $\mathcal{A}$ if it is in Φ, and false, otherwise. A candidate world view $\mathcal{A}$ w.r.t. a guess Φ is defined to be a *world view* under our approach if Φ is maximal, i.e., there is no other candidate world view $\mathcal{A}'$ w.r.t. a guess $\Phi' \supset \Phi$. Note that it is by applying a *maximal guess* Φ that we realize knowledge minimization with epistemic negation. Obviously, our definitions of program transformations and world views fundamentally differ from those of Gelfond [13,14], and Kahl et al. [19,18].

(3) The approaches of Gelfond [13,14] are said to suffer from the problem with recursion through **M** because they yield for the logic program $\Pi = \{p \leftarrow \mathbf{M}p\}$ two world views, viz. $\mathcal{A}_1 = \{\{p\}\}$ and $\mathcal{A}_2 = \{\emptyset\}$; however, although Kahl et al. [18] presented a preference order over $\mathbf{K}p$, p and $\mathbf{M}p$ as an intuitive justification, to the best of our knowledge there has been no deeper discussion or principled justification in the literature on why $\mathcal{A}_1$ is the right world view of this program and $\mathcal{A}_2$ is not. In fact, there has been no formal characterization of the problem with recursion through **M** in the literature [14,33,19,18,4]. In this paper, we provide a formal definition of this problem in terms of knowledge minimization with epistemic negation and show that our approach to evaluating epistemic negation is free of both the problem of unintended world views due to recursion through **K** and the problem due to **M**.

(4) The proposed approach to evaluating epistemic negation can be used to extend any existing answer set semantics with epistemic negation, such as those semantics defined in [26,27,32,1,8,12,29]. For illustration, we extend as a simple showcase the well-known FLP semantics of Faber et al. [8], yielding a new semantics called epistemic FLP semantics. We also extend the more restrictive well-justified FLP semantics of Shen et al. [29], which is free of circularity for default negation, to an epistemic well-justified semantics.

(5) As satisfiability of an arbitrary general logic program is undecidable, we address the computational complexity of epistemic FLP semantics for propositional programs. In particular, we show that deciding whether a propositional program Π has epistemic FLP answer sets is Σ_3^p-complete. Furthermore, we show that query evaluation, i.e., deciding whether a propositional formula is true in every epistemic FLP answer set of some world view of Π, is Σ_4^p-complete in general; important fragments, e.g. programs that match normal epistemic specifications, have lower complexity.

The paper is organized as follows. In Section 2 we introduce a first-order logic language and define logic programs with first-order formulas and epistemic negation. In Section 3 we present a novel program transformation and define epistemic reducts. In Section 4 we present a general framework for defining epistemic answer set semantics. As a simple showcase of this general semantics, in Section 5 we extend FLP answer set semantics with epistemic negation and study in depth the computational complexity of the extended semantics. In Section 6 we extend the well-justified FLP semantics with epistemic negation. In Section 7 we discuss related work, and in Section 8 we conclude with some future work.

In order not to distract from the flow of reading, proofs of theorems are in the appendix.

2. Preliminaries

In this section, we introduce a first-order logic language and define logic programs with first-order formulas and epistemic negation.

In the sequel, for a set S, $|S|$ denotes the number of elements in S; for two sets I and J, $I \subseteq J$ or $J \supseteq I$ denotes I is a subset of J; in particular, $I \subset J$ or $J \supset I$ denotes I is a proper subset of J.

2.1. A first-order logic language

We follow the notation in [29] and define a first-order logic language $\mathcal{L}_\Sigma$ with equality over a *signature* $\Sigma = (\mathcal{P}, \mathcal{F})$, where $\mathcal{P}$ and $\mathcal{F}$ are countable sets of *predicate* and *function* symbols, respectively; $\mathcal{C} \subseteq \mathcal{F}$ denotes the set of 0-ary function symbols, which are called *constants*. *Variables, terms, atoms* and *literals* are defined as usual. We denote variables with strings starting with X, Y or Z.

First-order formulas (briefly formulas) are constructed as usual from atoms using connectives $\neg$, $\wedge$, $\vee$, $\supset$, $\top$, $\bot$, $\exists$ and $\forall$, where $\top$ and $\bot$ are two 0-place logical connectives expressing *true* and *false*, respectively. Formulas are *closed* if they contain no free variables, i.e., each variable occurrence is in the scope of some quantifier. A *first-order theory* (or theory) is a set of closed formulas. Terms, atoms and formulas are *ground* if they have no variables, and *propositional* if they contain no variables, no function symbols except constants, and no equalities. By $\mathcal{N}_\Sigma$ we denote the set of all ground terms of Σ, and by $\mathcal{H}_\Sigma$ the set of all ground atoms.

In this paper we consider *SNA interpretations*, i.e., interpretations which employ the well-known *standard names assumption* (SNA) [3,24]. An SNA interpretation (or interpretation for short) I of $\mathcal{L}_\Sigma$ is a subset of $\mathcal{H}_\Sigma$ such that for any ground atom A, I *satisfies* A if $A \in I$, and I satisfies $\neg A$ if $A \notin I$. The notion of *satisfaction/models* of a formula/theory in I is defined as usual. A theory T is *consistent* or *satisfiable* if T has a model. We say that T *entails* a closed formula F, denoted $T \models F$, if all models of T are models of F. Furthermore F is *true* (resp. *false*) in an interpretation I if I satisfies (resp. does not satisfy) F.

2.2. Logic programs with epistemic negation

Based on the first-order logic language $\mathcal{L}_\Sigma$ defined above, we introduce the syntax of a logic program with epistemic negation and define its grounding, satisfaction and models.

2.2.1. Syntax

As in [29] and early AI literature, default negation will be expressed by giving the connective $\neg$ a special meaning; i.e., we use $\neg$ to denote the default negation operator. We further extend the language $\mathcal{L}_\Sigma$ to include the epistemic negation operator **not** and the rule operator $\leftarrow$.

Epistemic formulas are constructed from atoms using the connectives $\neg, \wedge, \vee, \supset, \top, \bot, \exists, \forall$ together with the operator **not** in the same way as first-order formulas. An *epistemic negation* is an epistemic formula of the form **not** F, where F is an epistemic formula; it is *non-nested* if F contains no epistemic negation, and *nested*, otherwise. An epistemic formula is *closed* if it contains no free variables.

Let E be an epistemic formula and E' be E with every free variable replaced by a constant (this process is called *grounding*). E is *instance-closed* (w.r.t. epistemic negation) if for every epistemic negation **not** F in E', F itself is a closed epistemic formula. For instance, $person(X) \wedge \mathbf{not}\, guilty(X)$ is an instance-closed epistemic formula, as when the free variable X is replaced by a constant, the epistemic negation $\mathbf{not}\, guilty(X)$ will become closed; however, $\forall X(person(X) \wedge \mathbf{not}\, guilty(X))$ is not instance-closed because X is not a free variable and the epistemic negation $\mathbf{not}\, guilty(X)$ will not become closed after grounding.

We use instance-closed epistemic formulas to construct rules and logic programs.

Definition 1 *(General logic program).* A *general logic program* (logic program for short) is a finite set of *rules* of the form $H \leftarrow B$, where H and B are instance-closed epistemic formulas without nested epistemic negations.

Note that like the approaches of Gelfond [14] and Kahl et al. [18], where modal operators **K** and **M** are not nested in logic programs, we do not consider logic programs with nested epistemic negations (**not** is also a modal operator); a nested epistemic negation like **not**(**not** F) intuitively expresses "That F cannot be proved to be true cannot be proved to be true" and such expressions seem to have rare applications in practical scenarios.

For convenience, for a rule $r: \ H \leftarrow B$ we refer to B and H as the body and head of r, denoted $body(r)$ and $head(r)$, respectively. When $head(r)$ is empty, we rewrite the rule as $\bot \leftarrow body(r)$; when $body(r)$ is empty, we omit the rule operator $\leftarrow$.

Definition 2 *(Normal epistemic program).* A *normal epistemic program* is a logic program consisting of rules of the form

$$A_0 \leftarrow A_1 \wedge \ldots \wedge A_m \wedge \mathbf{not}\, A_{m+1} \wedge \ldots \wedge \mathbf{not}\, A_n \tag{4}$$

where $n \geq m \geq 0$ and each A_i is an atom without equality and function symbols except constants.

Furthermore, as usual a *normal logic program* consists of rules of the above form (4) except that epistemic negation **not** is replaced by default negation $\neg$; a *positive logic program* is a $\neg$-free normal logic program.

Definition 3 *(Propositional program).* A *propositional program* Π is a logic program which contains no variables, no function symbols except constants, and no equalities. The *Herbrand base* of Π is defined as usual. Any subset of the Herbrand base is a *Herbrand interpretation* of Π.

2.2.2. Grounding

In a logic program Π, some rules may contain free variables. In ASP, these free variables will be instantiated by constants from a finite set – usually the set $\mathcal{C}_\Pi$ of constants occurring in Π. Without loss of generality, we assume that $\mathcal{C}_\Pi$ consists of all constants in Π (in case that some constant a of the domain does not appear in Π, we may have it by adding to Π a dummy rule $p(a) \leftarrow p(a)$). Then for any logic program Π, $\mathcal{C}_\Pi$ is unique.

A *closed instance* of a rule is the rule with all free variables replaced by constants in $\mathcal{C}_\Pi$. The *grounding* of Π, denoted $ground(\Pi)$, is the set of all closed instances of all rules in Π.

Note that each rule $H \leftarrow B$ with the set S of free variables may also be viewed as a *globally* universally quantified rule $\forall S(H \leftarrow B)$, where the domain of each variable in S is $\mathcal{C}_\Pi$ while the domain of the other (*locally* quantified) variables is $\mathcal{N}_\Sigma$. Only globally universally quantified variables will be instantiated over their domain $\mathcal{C}_\Pi$ for the grounding $ground(\Pi)$.

To sum up, a logic program Π is viewed as shorthand for $ground(\Pi)$, where each free variable in Π is viewed as shorthand for constants in $\mathcal{C}_\Pi$.

As rule bodies and heads in Π are instance-closed epistemic formulas, for every epistemic negation **not** F in $ground(\Pi)$, F itself is a closed epistemic formula. Therefore, in the sequel unless otherwise stated, for any epistemic negation **not** F we assume F is a closed epistemic formula.

2.2.3. Satisfaction and models

Next, we extend the satisfaction relation of $\mathcal{L}_\Sigma$ to logic programs. As epistemic negation works at a meta level over a collection of interpretations, the definition of satisfaction/models of epistemic formulas should be based on a collection of interpretations.

Definition 4. Let $\mathcal{A}$ be a collection of interpretations and $I \in \mathcal{A}$.

(1) Let F be a closed formula. Then **not** F is true in $\mathcal{A}$ (or $\mathcal{A}$ satisfies **not** F) if F is false in some $J \in \mathcal{A}$, and false, otherwise. Furthermore I satisfies **not** F w.r.t. $\mathcal{A}$ if **not** F is true in $\mathcal{A}$.
(2) I satisfies a closed epistemic formula E w.r.t. $\mathcal{A}$ if I satisfies E as in first-order logic except that the satisfaction of epistemic negations in E is determined by (1).
(3) I satisfies a closed instance r of a rule w.r.t. $\mathcal{A}$ if I satisfies $head(r)$ w.r.t. $\mathcal{A}$ once I satisfies $body(r)$ w.r.t. $\mathcal{A}$.
(4) $\mathcal{A}$ is a *collection of models* of a logic program Π if every $J \in \mathcal{A}$ satisfies all rules in $ground(\Pi)$ w.r.t. $\mathcal{A}$. In this case, every $J \in \mathcal{A}$ is a model of Π (w.r.t. $\mathcal{A}$). The program Π is *consistent* if it has a model.

Observe the following properties of satisfaction of a logic program Π.

First, when Π contains no epistemic negation, satisfaction in Definition 4 reduces to that in first-order logic. In this case, we omit $\mathcal{A}$. Then I satisfies a closed rule instance r if I satisfies $head(r)$ or I does not satisfy $body(r)$, and I is a model of Π if it satisfies all rules in $ground(\Pi)$; moreover, a model I is *minimal* if Π has no model J that is a proper subset of I.

Second, satisfaction of an epistemic negation **not** F in I w.r.t. $\mathcal{A}$ is determined only by $\mathcal{A}$, i.e., I can be ignored.

Finally, when $\mathcal{A}$ consists of a single interpretation I, the notions of satisfaction, models and consistency are the same as in first-order logic, i.e., they are determined only by I and $\mathcal{A}$ can be ignored. In this special case, the epistemic negation operator **not** coincides with the operator $\neg$, i.e., I satisfies **not** F w.r.t. $\mathcal{A}$ iff I satisfies $\neg F$ iff F is false in I, where F is a closed formula. Hence the following proposition is immediate.

Proposition 1. *Let Π be a logic program and $\Pi^\neg$ be Π with all epistemic negations* **not** F *replaced by default negations $\neg F$. For any interpretation I, $\mathcal{A} = \{I\}$ is a collection of models of Π iff I is a model of $\Pi^\neg$.*

The following theorem is important, as it lays a theoretical basis for the introduction of our novel program transformation, which is described in the next section.

Theorem 1. *Let Π be a logic program such that for every* **not** F *in $ground(\Pi)$, F is true in every model of Π. Let $\Pi^\neg$ be Π with each epistemic negation* **not** F *replaced by default negation $\neg F$. Then Π and $\Pi^\neg$ have the same models.*

3. Program transformation and epistemic reducts

In ASP, it is common to transform a logic program into a reduct which is free of negation or modal operators. For instance, for a normal logic program Π, the seminal *GL-reduct* Π^I w.r.t. a given interpretation I is obtained from $ground(\Pi)$ by removing first all rules whose bodies contain a default negation $\neg A$ with $A \in I$, and then all $\neg A$ from the remaining rules [16]. Similarly, when Π is a logic program extended with modal operators **K** and **M**, Gelfond [13,14], Truszczynski [33] and Kahl [19] defined a transformation w.r.t. a given set $\mathcal{A}$ of interpretations by eliminating/replacing all modal literals in $ground(\Pi)$ in terms of whether or not they are true in $\mathcal{A}$.

Note that these existing definitions of program transformations are based on an assumption, which is either a given interpretation or a given set of interpretations, and default negations or modal literals in a logic program are evaluated against the assumption.

In this paper we aim to apply epistemic negation to minimize the knowledge in a world view of a general logic program Π by assuming every epistemic negation **not** F in Π to be true in the world view whenever possible. To this end, we define program transformations in an alternative way, which is based on an assumption that is *a given set of epistemic negations*, instead of a given set of interpretations.

Definition 5. For a logic program Π, let $Ep(\Pi)$ denote the set of all epistemic negations **not** F in $ground(\Pi)$. A *guess of epistemic negations* for Π is a subset Φ of $Ep(\Pi)$.

Intuitively for every **not** $F \in \Phi$, it is guessed that F couldn't be proved true, and for every **not** $F \in Ep(\Pi) \setminus \Phi$, it is guessed that F would be proved true. Recall that an epistemic negation **not** F expresses that there is no evidence proving that F is true, where F is *proved true* if it is true in every answer set of some world view.

Once a guess Φ is given, we can transform program Π by replacing all epistemic negations in terms of Φ. There would be different replacements for epistemic negations, which would lead to different program transformations. The simplest yet naive one is to replace **not** F with $\top$ if **not** $F \in \Phi$, and with $\bot$, otherwise. It turns out that this transformation incurs

both the problem of unintended world views due to recursion through **K** and the problem due to recursion through **M**, analogously to the cases in [13].

The key idea of our program transformation is as follows. We first assume that the guess on all **not** $F \in \Phi$ is correct and thus replace them with $\top$. Then, for every **not** $F \in Ep(\Pi) \setminus \Phi$, instead of replacing it with $\bot$, we replace it with $\neg F$. The intuition and rationale for the latter replacement is as follows: if Φ is a *correct guess*, once all epistemic negations **not** $F \in \Phi$ in *ground*(Π) are replaced with $\top$, which leads to a new program $\Pi^{\top}$, for every **not** F in $\Pi^{\top}$, the formula F is supposed to be true in every answer set of $\Pi^{\top}$. Let Π^{Φ} be $\Pi^{\top}$ with each **not** F replaced by $\neg F$; then by Theorem 1, where *model* is analogously replaced by *answer set*, we expect that $\Pi^{\top}$ and Π^{Φ} have the same answer sets. This rational justification of the replacements for epistemic negations leads to the following novel program transformation.

Definition 6 *(Epistemic reducts).* Let Π be a logic program and let $\Phi \subseteq EP(\Pi)$ be a guess of epistemic negations for Π. The *epistemic reduct* Π^{Φ} of Π w.r.t. Φ is obtained from *ground*(Π) by replacing every **not** $F \in \Phi$ with $\top$, and every **not** $F \in Ep(\Pi) \setminus \Phi$ with $\neg F$. We call Π *consistent* w.r.t. Φ if Π^{Φ} is consistent.

In the Introduction we mentioned that a world view $\mathcal{A}$ is said to have an epistemic circular justification if some object literal L being true in some interpretation $I \in \mathcal{A}$ is due to $\mathbf{K}L$ (or its equivalent modal literals expressing that L is true in every interpretation $J \in \mathcal{A}$) being treated true in the program transformation w.r.t. $\mathcal{A}$. In our language, $\mathbf{K}L$ is shorthand for $\neg$ **not** L, and in our program transformation w.r.t. a guess Φ, $\neg$ **not** L will be either treated $\neg\top$ (when **not** $L \in \Phi$), which evaluates to false, or treated $\neg\neg L$ (when **not** $F \in Ep(\Pi) \setminus \Phi$), which evaluates to L. This means that our definition of the program transformation would never incur epistemic circular justifications and thus guarantees that world views based on the epistemic reducts will be free of the problem with recursion through **K**.

4. A general epistemic answer set semantics

Now that all epistemic negations have been removed from a logic program Π, leading to an epistemic reduct Π^{Φ} w.r.t. a guess Φ, we can apply any answer set semantics for logic programs without epistemic negation to compute all answer sets $\mathcal{A}$ of Π^{Φ}. For $\mathcal{A}$ to be a world view, it must agree with the guess Φ, i.e., every **not** $F \in \Phi$ is true and every **not** $F \in Ep(\Pi) \setminus \Phi$ is false in $\mathcal{A}$; it should also satisfy the property of knowledge minimization with epistemic negation, as defined below.

Definition 7. A world view $\mathcal{A}$ of a logic program Π is said to have the *property of knowledge minimization with epistemic negation* if $\mathcal{A}$ satisfies a maximal set Φ of epistemic negations in $Ep(\Pi)$ (i.e., no other world view satisfies a set $\Phi' \supset \Phi$ of epistemic negations in $Ep(\Pi)$).

In this section, we present a general framework for defining epistemic answer set semantics, thus called a *general epistemic answer set semantics*, which is applicable to extend any existing answer set semantics with epistemic negation.

Definition 8 *(General epistemic semantics).* Let Π be a logic program and let Φ be a guess such that Π^{Φ} is a consistent epistemic reduct. Let $\mathcal{X}$ be an answer set semantics for logic programs without epistemic negation. The collection $\mathcal{A}$ of all answer sets of Π^{Φ} under $\mathcal{X}$ is a *candidate world view* of Π w.r.t. Φ if (a) $\mathcal{A} \neq \emptyset$, (b) every **not** $F \in \Phi$ is true in $\mathcal{A}$, and (c) every **not** $F \in Ep(\Pi) \setminus \Phi$ is false in $\mathcal{A}$. A candidate world view $\mathcal{A}$ w.r.t. a guess Φ is a *world view* if Φ is maximal (i.e., there is no other candidate world view w.r.t. a guess $\Phi' \supset \Phi$).

The condition "Φ is maximal" implies that world views under the general epistemic semantics have the property of knowledge minimization with epistemic negation.

In this paper, when we say that a formula F is true in a logic program Π, we mean that F is proved true in Π. Recall that a formula is proved true if it is true in every answer set of some world view. Therefore, we define the following form of query evaluation.[7]

Definition 9 *(Query evaluation).* Let Π be a logic program and F a closed formula. We say F is true in Π under the general epistemic semantics if Π has a world view $\mathcal{A}$ such that F is true in every answer set in $\mathcal{A}$.

Now we are ready to introduce the following formal definition of the problem of unintended world views with recursion through **M**, which was informally described in the Introduction.

Definition 10 *(The problem with recursion through **M**).* An epistemic answer set semantics is said to have the *problem of unintended world views due to recursion through* **M** if its world views do not satisfy the property of knowledge minimization with epistemic negation.

[7] Depending on different applications, one could explore other forms of query evaluation, such as: a closed formula F is true if it is true in every answer set of every world view.

The above formulation shows that in essence, an epistemic answer set semantics with the problem with recursion through **M** fails to make *full use* of the epistemic negations **not** F occurring in a logic program to minimize the knowledge by allowing these epistemic negations to be true in world views whenever possible. This will be further illustrated in Example 6.

Obviously, our general epistemic answer set semantics is free of the problem with recursion through **M**.

Remark 1. We stress that default negation and epistemic negation are used to minimize the knowledge at the *answer set* level and the *world view* level, respectively, and the notion of *positive knowledge* to be minimized at the answer set level is different from that at the world view level. At the answer set level, where default negation $\neg$ is used, any ground atom is the positive knowledge to be minimized in answer sets; but at the world view level, where epistemic negation **not** is used, any closed (first-order) formula is the positive knowledge to be minimized in world views. Specifically, at the answer set level, for any ground atom A we assume its default negation $\neg A$ to be true in every answer set whenever possible (*knowledge minimization with default negation*); analogously at the world view level, for any closed formula F we assume its epistemic negation **not** F occurring in a logic program to be true in every world view whenever possible (*knowledge minimization with epistemic negation*). As a result, by applying default negation we minimize the knowledge in every answer set, and by applying epistemic negation we minimize the knowledge in every world view. Since epistemic negation is at a meta level, in our approach the minimization with epistemic negation has higher priority and is done before the minimization with default negation.

Note that if one intends to apply epistemic negation to a formula F by assuming **not** F to be true in every world view whenever possible, one must *explicitly* express the epistemic negation **not** F in a logic program. Thus the following four programs

$$\Pi_1 = \{p \vee q\}$$
$$\Pi_2 = \{p \vee q,\ p \leftarrow \mathbf{not}\, q\}$$
$$\Pi_3 = \{p \vee q,\ q \leftarrow \mathbf{not}\, p\}$$
$$\Pi_4 = \{p \vee q,\ p \leftarrow \mathbf{not}\, q,\ q \leftarrow \mathbf{not}\, p\}$$

are entirely different and have different world views: Π_1 has a unique world view $\{\{p\}\{q\}\}$, Π_2 has a unique one $\{\{p\}\}$, Π_3 has a unique one $\{\{q\}\}$, and Π_4 has two world views $\{\{p\}\}$ and $\{\{q\}\}$.

In contrast, for any ground atom A its default negation $\neg A$ is *implicitly* assumed to be true in every answer set whenever possible, whether or not $\neg A$ is present in a logic program. Thus the following four programs

$$\Pi_1 = \{p \vee q\}$$
$$\Pi_2 = \{p \vee q,\ p \leftarrow \neg q\}$$
$$\Pi_3 = \{p \vee q,\ q \leftarrow \neg p\}$$
$$\Pi_4 = \{p \vee q,\ p \leftarrow \neg q,\ q \leftarrow \neg p\}$$

have the same answer sets $\{p\}$ and $\{q\}$ under the existing answer set semantics defined in [17,26,27,32,1,8,12,29].

5. FLP answer set semantics with epistemic negation

The general framework of Definition 8 is applicable to extend any existing answer set semantics, such as those defined in [26,27,32,1,8,12,29], with epistemic negation; as a simple showcase, we extend in this section the well-known FLP answer set semantics of Faber et al. [8] to logic programs with epistemic negation. The FLP semantics coincides with the stable model respectively answer set semantics for normal and disjunctive logic programs in [16,17], and can be regarded as a base semantics for extensions of logic programs beyound aggregates [6]. The following definition is from Shen et al. [29], which lifts Faber et al.'s FLP semantics to general logic programs without epistemic negation.

Definition 11. Let Π be a logic program without epistemic negation and I an interpretation. The *FLP-reduct* of Π w.r.t. I is $f\Pi^I = \{r \in ground(\Pi) \mid I \text{ satisfies } body(r)\}$, and I is an *FLP answer set* of Π if I is a minimal model of $f\Pi^I$.

By replacing $\mathcal{X}$ with FLP in Definition 8, we obtain an epistemic FLP answer set semantics (EFLP semantics for short).

Definition 12 *(EFLP semantics).* Let Π be a logic program and Φ a guess such that Π^Φ is a consistent epistemic reduct. The collection $\mathcal{A}$ of all FLP answer sets of Π^Φ is a *candidate world view*, resp. a *world view*, of Π w.r.t. Φ if $\mathcal{A}$ satisfies the conditions of Definition 8. Every FLP answer set in a world view is called an *EFLP answer set*.

Obviously, for logic programs without epistemic negation, $Ep(\Pi) = \emptyset$ and EFLP semantics reduces to FLP semantics.

The following result shows that EFLP answer sets of Π are models of Π.

Theorem 2. *Let Π be a logic program and let $\mathcal{A}$ be a world view of Π w.r.t. a guess Φ under EFLP semantics. Then $\mathcal{A}$ is a collection of models of Π.*

Let Π be a normal epistemic program and $\Pi^{\neg}$ be the normal logic program obtained from Π by replacing epistemic negation **not** with default negation $\neg$. Let I be an interpretation and $(\Pi^{\neg})^I$ be the GL-reduct of $\Pi^{\neg}$; then I is a *standard answer set* of $\Pi^{\neg}$ if I is the least model of $(\Pi^{\neg})^I$ [16].

The following result shows that answer sets of a normal epistemic program Π under EFLP semantics coincide with those of $\Pi^{\neg}$ under the standard answer set semantics.

Theorem 3. *Let Π be a normal epistemic program and let $\Pi^{\neg}$ be the normal logic program obtained from Π by replacing* **not** *with $\neg$. Then (1) every world view $\mathcal{A}$ of Π fulfills $|\mathcal{A}| = 1$ (i.e., $\mathcal{A}$ is a singleton), and (2) $\mathcal{A} = \{I\}$ is a world view of Π iff I is an FLP answer set of $\Pi^{\neg}$ iff I is a standard answer set of $\Pi^{\neg}$.*

Next we consider a few simple examples to illustrate the novelty and suitability of our approach. Interested readers may apply the approach to more examples collected in [19,18].

Example 2. The following logic program uses epistemic negation to formalize the well-known presumption of innocence:

$$\begin{array}{lll} \Pi_1\colon & \mathit{innocent}(\mathit{John}) \vee \mathit{guilty}(\mathit{John}) & r_1 \\ & \mathit{innocent}(X) \leftarrow \mathbf{not}\, \mathit{guilty}(X) & r_2 \end{array}$$

Rule r_1 says that *John* is either innocent or guilty, and rule r_2 asserts that one is presumed innocent if there is no evidence proving s/he is guilty. From this program we intend to arrive at the conclusion that *John* is innocent; specifically, $\mathit{innocent}(\mathit{John})$ is expected to be in every answer set of every world view of Π_1.

The grounding of Π_1 is

$$\begin{array}{lll} \mathit{ground}(\Pi_1)\colon & \mathit{innocent}(\mathit{John}) \vee \mathit{guilty}(\mathit{John}) & r_1 \\ & \mathit{innocent}(\mathit{John}) \leftarrow \mathbf{not}\, \mathit{guilty}(\mathit{John}) & r_2' \end{array}$$

It contains only one epistemic negation, thus $\mathit{Ep}(\Pi_1) = \{\mathbf{not}\, \mathit{guilty}(\mathit{John})\}$. So we have two guesses: $\Phi_1 = \{\mathbf{not}\, \mathit{guilty}(\mathit{John})\}$ and $\Phi_2 = \emptyset$.

We start with the largest guess Φ_1 and check if there is a world view w.r.t. Φ_1. Note $\mathit{Ep}(\Pi_1) \setminus \Phi_1 = \emptyset$. The epistemic reduct w.r.t. Φ_1 is

$$\begin{array}{lll} \Pi_1^{\Phi_1}\colon & \mathit{innocent}(\mathit{John}) \vee \mathit{guilty}(\mathit{John}) & r_1 \\ & \mathit{innocent}(\mathit{John}) \leftarrow \top & r_2'' \end{array}$$

$\Pi_1^{\Phi_1}$ is consistent and has a unique FLP answer set $I = \{\mathit{innocent}(\mathit{John})\}$. Let $\mathcal{A} = \{I\}$. As $\mathit{guilty}(\mathit{John})$ is false in I, $\mathbf{not}\, \mathit{guilty}(\mathit{John})$ is true in $\mathcal{A}$, thus $\mathcal{A}$ is a candidate world view of Π_1. Since Φ_1 is the largest guess, $\mathcal{A}$ is a world view of Π_1 w.r.t. Φ_1.

As $\Phi_2 \subset \Phi_1$, there would be no world view w.r.t. Φ_2. Hence, Π_1 has only one world view $\mathcal{A} = \{\{\mathit{innocent}(\mathit{John})\}\}$. Therefore, the formula $\mathit{innocent}(\mathit{John})$ is true in Π_1, meaning that *John* is innocent. This conforms to our expectation.

Example 3. To demonstrate the necessity of epistemic reasoning with epistemic negation, Gelfond [13] introduced the well-known college scholarship awarding problem with the following logic program:

$$\begin{array}{lll} \Pi_2\colon & \mathit{eligible}(X) \leftarrow \mathit{highGPA}(X) & r_1 \\ & \mathit{eligible}(X) \leftarrow \mathit{minority}(X) \wedge \mathit{fairGPA}(X) & r_2 \\ & \sim \mathit{eligible}(X) \leftarrow \mathit{lowGPA}(X) & r_3 \\ & \mathit{interview}(X) \leftarrow \neg \mathit{eligible}(X) \wedge \neg\sim \mathit{eligible}(X) & r_4 \\ & \mathit{fairGPA}(\mathit{Mike}) \vee \mathit{highGPA}(\mathit{Mike}) & r_5 \end{array}$$

Gelfond argued that *Mike* was intended to be interviewed, i.e., $\mathit{interview}(\mathit{Mike})$ should be included in every answer set of Π_2; however, he observed that the fourth rule r_4 was not powerful enough to formalize the intended statement that the students whose eligibility is not determined by the college rules should be interviewed by the scholarship committee. It was due to this observation that Gelfond [13,14] proposed his modal formalism in which r_4 was replaced by the rule

$$\mathit{interview}(X) \leftarrow \neg\mathbf{K}\, \mathit{eligible}(X) \wedge \neg\mathbf{K}\sim \mathit{eligible}(X).$$

Next, we show that this problem can be suitably handled using epistemic negation under EFLP semantics. As shown in [17], for any atom $p(X)$, the strong negation $\sim p(X)$ can be compiled away by introducing a fresh predicate $p'(X)$ along with a rule $\bot \leftarrow p(X) \wedge p'(X)$. We formulate the above problem by rewriting Π_2 by replacing $\neg$ with **not** and $\sim \mathit{eligible}(X)$ with $\mathit{eligible}'(X)$, and adding the rule $\bot \leftarrow \mathit{eligible}(X) \wedge \mathit{eligible}'(X)$; this yields a new program Π_2' with the grounding

$$
\begin{array}{lll}
ground(\Pi_2'): & eligible(Mike) \leftarrow highGPA(Mike) & r_1' \\
& eligible(Mike) \leftarrow minority(Mike) \wedge fairGPA(Mike) & r_2' \\
& eligible'(Mike) \leftarrow lowGPA(Mike) & r_3' \\
& interview(Mike) \leftarrow \mathbf{not}\, eligible(Mike) \wedge \mathbf{not}\, eligible'(Mike) & r_4' \\
& fairGPA(Mike) \vee highGPA(Mike) & r_5 \\
& \bot \leftarrow eligible(Mike) \wedge eligible'(Mike) & r_6
\end{array}
$$

$Ep(\Pi_2') = \{\mathbf{not}\, eligible(Mike),\ \mathbf{not}\, eligible'(Mike)\}$, so there are four guesses. We start with the largest guess $\Phi_1 = \{\mathbf{not}\, eligible(Mike), \mathbf{not}\, eligible'(Mike)\}$ and check if Π_2' has a world view w.r.t. Φ_1.

The epistemic reduct ${\Pi_2'}^{\Phi_1}$ w.r.t. Φ_1 is $ground(\Pi_2')$ except that r_4' is replaced by the rule

$$interview(Mike) \leftarrow \top \wedge \top \qquad r_4''$$

${\Pi_2'}^{\Phi_1}$ is consistent and has the following two FLP answer sets:

$$I_1 = \{fairGPA(Mike), interview(Mike)\}$$
$$I_2 = \{highGPA(Mike),\ eligible(Mike), interview(Mike)\}$$

Let $\mathcal{A} = \{I_1, I_2\}$. As $eligible(Mike)$ and $eligible'(Mike)$ are both false in I_1, both $\mathbf{not}\, eligible(Mike)$ and $\mathbf{not}\, eligible'(Mike)$ are true in $\mathcal{A}$. As $Ep(\Pi_2') \setminus \Phi_1 = \emptyset$ and Φ_1 is the largest guess, $\mathcal{A}$ is a unique world view of Π_2'.

Note that $interview(Mike)$ appears in every answer set in $\mathcal{A}$ and thus is true in this logic program. Therefore, *Mike* should be interviewed, as we expected.

Remark 2. We can also formulate the above problem without using strong negation by rewriting Π_2 with $\neg$ replaced by **not**, and $\sim$ by $\neg$; thus r_4 is rewritten as

$$interview(X) \leftarrow \mathbf{not}\, eligible(X) \wedge \mathbf{not}\, \neg eligible(X).$$

This rule directly expresses that a student should be interviewed if we can neither prove s/he is eligible nor not eligible. Then, the grounding of the new program consists of r_1', r_2', r_5, and the following two rules:

$$
\begin{array}{ll}
\neg eligible(Mike) \leftarrow lowGPA(Mike) & \overline{r_3} \\
interview(Mike) \leftarrow \mathbf{not}\, eligible(Mike) \wedge \mathbf{not}\, \neg eligible(Mike) & \overline{r_4}
\end{array}
$$

It is easy to check that this new program has a unique world view $\mathcal{A} = \{I_1, I_2\}$ w.r.t. the largest guess $\Phi_1 = \{\mathbf{not}\, eligible(Mike), \mathbf{not}\, \neg eligible(Mike)\}$, where

$$I_1 = \{fairGPA(Mike),\ interview(Mike)\}$$
$$I_2 = \{highGPA(Mike),\ eligible(Mike), interview(Mike)\}$$

which shows that *Mike* should be interviewed, as expected.

Example 4 *(Closed world rules).* Under EFLP semantics, we can directly formulate the closed world assumption using closed world rules of the form $\neg p \leftarrow \mathbf{not}\, p$, which expresses that when failing to prove p to be true, we assert $\neg p$. Moreover, we can also state its opposite using rules of the form $p \leftarrow \mathbf{not}\, \neg p$, which expresses that when failing to prove $\neg p$ to be true, we assert p. We can further combine them, leading to the following interesting logic program.

$$
\begin{array}{lll}
\Pi: & \neg p \leftarrow \mathbf{not}\, p & r_1 \\
& p \leftarrow \mathbf{not}\, \neg p & r_2
\end{array}
$$

It is easy to check that this program has two world views: $\mathcal{A}_1 = \{\emptyset\}$ w.r.t. the guess $\Phi_1 = \{\mathbf{not}\, p\}$ and $\mathcal{A}_2 = \{\{p\}\}$ w.r.t. $\Phi_2 = \{\mathbf{not}\, \neg p\}$. This conforms to our intuition that either $\neg p$ or p can be concluded from Π, depending on whether we choose to apply CWA on p (rule r_1) or on $\neg p$ (rule r_2).

Remark 3. Gelfond and Lifschitz [17] used a rule $\sim p \leftarrow \neg p$ (instead of $\neg p \leftarrow \mathbf{not}\, p$) to formalize CWA on an atom p, which achieves the effect that if p is not in an answer set I, then $\sim p$ is in I. As indicated by Gelfond [13], this formalization of CWA is problematic. For example, by this formalization the logic program $\Pi = \{p \vee q\}$ extended with the rule $\sim p \leftarrow \neg p$ would have two answer sets, viz. $I_1 = \{\sim p, q\}$ and $I_2 = \{p\}$; thus both p and q are *unknown* in Π under the semantics of

Gelfond and Lifschitz [17].[8] In contrast, if Π is extended with the CWA rule $\neg p \leftarrow \mathbf{not}\, p$, we would obtain a unique world view $\{\{q\}\}$ under EFLP semantics, so p is false and q is true in Π, as expected.[9]

Example 5. Let $\Pi = \{r \leftarrow \mathbf{not}\, p \wedge \neg r,\ p \leftarrow \neg q,\ q \leftarrow \neg p\}$. Intuitively, the last two rules say that either $\{p\}$ or $\{q\}$ is an answer set of Π, and the first rule is a *constraint* that every answer set of Π should contain p. Therefore, Π intuitively should have a unique answer set $\{p\}$. It is easy to check that under EFLP semantics this program has a unique world view $\mathcal{A} = \{\{p\}\}$ w.r.t. the guess $\Phi = \emptyset$. Note that $\mathcal{A}$ is not a world view under the approach of Gelfond [14], where $\mathbf{not}\, p$ in Π is replaced by $\neg \mathbf{K} p$.

Example 6. Consider again the logic program $\Pi = \{p \leftarrow \mathbf{M} p\}$. As mentioned earlier, $\mathbf{M} p$ is shorthand for $\mathbf{not} \neg p$, so this program can be rewritten as $\Pi = \{p \leftarrow \mathbf{not} \neg p\}$. The set $\Phi = \{\mathbf{not} \neg p\}$ is the largest guess and it is easy to check that $\{\{p\}\}$ is a unique world view of Π w.r.t. Φ under EFLP semantics.

Similarly, the logic program in Example 1 can be rewritten as

$$\begin{array}{lll} \Pi: & p \leftarrow \mathbf{not} \neg q \wedge \neg q & r_1 \\ & q \leftarrow \mathbf{not} \neg p \wedge \neg p & r_2 \end{array}$$

It has a unique world view $\mathcal{A} = \{\{p\}, \{q\}\}$ w.r.t. the guess $\Phi = \{\mathbf{not} \neg q, \mathbf{not} \neg p\}$. For this program, as shown in Example 1 applying the approaches of Gelfond [13,14], and Kahl et al. [19,18] will produce two world views, including the unintended one $\{\emptyset\}$.

We remark that our epistemic answer set semantics minimizes the knowledge in world views by requiring that every world view should satisfy as many epistemic negations in a logic program as possible. *It is this property of knowledge minimization with epistemic negation that provides a principled justification for the above two programs on why $\mathcal{A}_1 = \{\{p\}\}$ is the right world view and $\mathcal{A}_2 = \{\emptyset\}$ is not.*

Remark 4. A rule of the form

$$p \leftarrow \mathbf{not} \neg p \tag{5}$$

(or $p \leftarrow \mathbf{M} p$) expresses that if there is no evidence to prove $\neg p$ to be true, we conclude p; put another way, we admit p unless we prove its contrary. Observe that such rules are commonly used in our daily life and our approach provides a suitable solution to such epistemic specifications. As an interesting example, the well-known *Murphy's law*[10] says that *anything that can go wrong, will go wrong;* put another way, if something can go wrong and we cannot prove it will not go wrong, then it will go wrong. We can conveniently formalize this law by a rule

$$gowrongUnderMurphylaw(X) \leftarrow \mathbf{not} \neg gowrongUnderMurphylaw(X).$$

As another close example, our research group has a regular weekly meeting and our secretary *Cathy* reminds us every week by a *no-reply email* with the assumption that every member in the group will be present unless some notice of absence is available. This epistemic assumption can be suitably formalized by a rule of the form

$$present(X) \leftarrow \mathbf{not} \neg present(X),$$

i.e., any group member will be present unless we prove the contrary. Then for an individual member *John* without notice to *Cathy*, we would have a grounding

$$\Pi = \{present(John) \leftarrow \mathbf{not} \neg present(John)\}$$

which has a unique world view $\{\{present(John)\}\}$ under EFLP semantics, meaning that *John* will be present in the meeting.

Example 7. Consider the logic program $\Pi = \{p \leftarrow \mathbf{not}\, p \vee p\}$. Note that $\mathbf{not}\, p \vee p$ is a *tautology* in that for any collection $\mathcal{A}$ of interpretations, every $I \in \mathcal{A}$ satisfies $\mathbf{not}\, p \vee p$ w.r.t. $\mathcal{A}$. Then p can be inferred by applying the rule $p \leftarrow \mathbf{not}\, p \vee p$; thus Π is supposed to have a unique world view $\mathcal{A} = \{\{p\}\}$.

This program has two guesses: $\Phi_1 = \{\mathbf{not}\, p\}$ and $\Phi_2 = \emptyset$. It is easy to check that Π has no world view w.r.t. Φ_1, but has a world view $\mathcal{A} = \{\{p\}\}$ w.r.t. Φ_2.

Note that the approaches of Gelfond [13,14] and Kahl et al. [19,18] are not applicable to this program.

Observe that in our logic language $\mathcal{L}_\Sigma$ the formula $\neg p \vee p$ is also a tautology, and $p \leftarrow \neg p \vee p$ is not equivalent to $\{p \leftarrow \neg p,\ p \leftarrow p\}$. Under EFLP semantics the rule $p \leftarrow \mathbf{not}\, p \vee p$ is not equivalent to $\{p \leftarrow \mathbf{not}\, p,\ p \leftarrow p\}$.

---

[8] Under the semantics of Gelfond and Lifschitz [17], p is true in an answer set I if $p \in I$, false if $\sim p \in I$, and *unknown*, otherwise. p is true/false in a logic program Π if it is true/false in all answer sets of Π, and unknown, otherwise.

[9] Alternatively, one might want to extend Π with the rule $\sim p \leftarrow \mathbf{not}\, p$. Then $\sim p$ can be compiled away similarly as in Example 3. The resulting program $\Pi' = \{p \vee q,\ p' \leftarrow \mathbf{not}\, p,\ \bot \leftarrow p \wedge p'\}$ has a unique world view $\{\{q, p'\}\}$ under EFLP semantics, meaning p is false and q is true in Π.

[10] https://en.wikipedia.org/wiki/Murphy%27s_law.

5.1. Computational complexity

For a logic program Π, each guess Φ leads to at most one candidate world view w.r.t. Φ. As there are at most $2^{|Ep(\Pi)|}$ guesses (where $|Ep(\Pi)|$ is the number of epistemic negations in $Ep(\Pi)$), Π has at most $2^{|Ep(\Pi)|}$ candidate world views. In view of the fact that if Π has a world view w.r.t. Φ then it will have no world view w.r.t. any $\Phi' \supset \Phi$ or $\Phi' \subset \Phi$, the following result is immediate.

Proposition 2. *A logic program Π has at most $\binom{n}{\lfloor n/2 \rfloor}$ many world views under EFLP semantics, where $n = |Ep(\Pi)|$.*

Indeed, $\binom{n}{\lfloor n/2 \rfloor}$ is the maximal size of an antichain in the powerset lattice of a set of cardinality n [31]. In the two most extreme cases that Π has a world view w.r.t. Φ, where $\Phi = Ep(\Pi)$ or $\Phi = \emptyset$, Π has only one world view.

As the satisfiability of arbitrary first-order theories is undecidable, we only address the computational complexity of EFLP answer sets for propositional programs under Herbrand interpretations. Recall that in this case, deciding the existence of FLP answer sets of a given logic program Π without epistemic negation is Σ_2^p-complete [29], where the Σ_2^p-hardness is inherited from disjunctive logic programs [8,5].

We first consider the complexity of recognizing a suitable guess.

Theorem 4. *Given a propositional program Π and a guess Φ for it, deciding whether Π has a candidate world view w.r.t. Φ under EFLP semantics is D_2^p-complete.*

The class D_2^p consists of all problems $(P_!, P_2)$ whose instances are pairs (I_1, I_2) of instances I_1 of a problem P_1 in Σ_2^p and I_2 of a problem P_2 in Π_2^p, respectively.

Informally, we must check the conditions (a)–(c) of a candidate world view in Definition 8, where (a) establishes *a fortiori* also consistency of Π^Φ; as Π^Φ is constructible in polynomial time, we can solve (a) and (b) in Σ_2^p and (c) in Π_2^p, as deciding whether Π^Φ has some FLP answer set that satisfies (resp. does not satisfy) a formula F is in Σ_2^p; thus, the problem is in D_2^p. The D_2^p-hardness is shown by a reduction from deciding whether, given a pair (Π_1, Π_2) of propositional programs, both Π_1 has some FLP-answer set and Π_2 has no FLP answer set; this problem is D_2^p-complete. Informally, Π_2 is modified to a program Π_2' whose answer sets amount to those of Π_2 plus an extra answer set $\{A\}$, where A is a fresh atom; to this end, we add $\neg A$ in the rule bodies of Π_2 and guessing clauses $A \leftarrow \neg \bar{A}$ and $\bar{A} \leftarrow \neg A$, where $\bar{A}$ is another fresh atom. The program Π contains Π_1 and Π_2' and has a candidate world view w.r.t. $\Phi = \emptyset$, where $EP(\Pi) = \{\mathbf{not}\, A\}$, just if Π_1 has some FLP answer set and A is true in all FLP answer sets of Π_2' (i.e., Π_2 has no FLP answer set). (For more details, see the proof in Appendix A.)

From this result, we obtain that deciding program consistency under candidate world views is at the third level of the polynomial hierarchy. More precisely,

Theorem 5. *Given a propositional program Π, deciding whether Π has (i) some candidate world view and (ii) some world view, i.e., deciding EFLP answer set existence, are both Σ_3^p-complete.*

Indeed, a guess $\Phi \subseteq Ep(\Pi)$ such that Π has some candidate world view w.r.t. Φ can be verified in polynomial time with an oracle for D_2^p, and hence also with one for Σ_2^p. This places the problem in Σ_3^p. The matching Σ_3^p-hardness can be shown by a reduction from evaluating quantified Boolean formulas of the form $\exists Z \forall Y \exists Z \phi$. Informally, epistemic negations $\mathbf{not}\, X$, $\mathbf{not}\, \bar{X}$ are used to guess an assignment to the X atoms, and some other epistemic negation serves to check that for each assignment to the Y atoms, some assignment to Z makes ϕ true; this test for Y and Z is encoded to cautious reasoning from the FLP answer set of a disjunctive logic program, which is Π_2^p-complete [5].

Note that some world view exists iff some candidate world view exists; this justifies the second part of Theorem 5, and consistency checking under candidate world view and world view semantics are equally hard. On the other hand, under EFLP semantics formula evaluation is harder.

Theorem 6. *Given a propositional program Π and a propositional formula F, deciding whether F is true in Π under EFLP semantics (i) w.r.t. candidate world views is Σ_3^p-complete and (ii) w.r.t. world views is Σ_4^p-complete.*

While under candidate world views, it suffices to guess $\Phi \subseteq EP(\Pi)$ such that Π has a candidate world view $\mathcal{A}$ w.r.t. Φ and Π^Φ cautiously entails F (i.e., F is true in all answer sets of Π^Φ, which can be checked with an Σ_2^p oracle), in case of world views under EFLP semantics, in addition maximality of Φ must be tested, i.e., no $\Phi' \supset \Phi$ has some candidate world view; this however turns out to be Π_3^p-complete, which lifts the problem to the fourth level of the polynomial hierarchy.

We note that the above results hold for propositional programs that amount to epistemic specifications, i.e., the rules match the syntax of form (1), where $\mathbf{K}F$ amounts to $\neg\, \mathbf{not}\, F$ and $\mathbf{M}F$ to $\mathbf{not} \neg F$. In case of normal epistemic specifications, i.e., the rules of form (1) with $m = 1$, the complexity drops by one level of the polynomial hierarchy, and we obtain D_1^p-, Σ_2^p and Σ_3^p-completeness in place of D_2^p-, Σ_3^p and Σ_4^p-completeness, respectively (for details, see Appendix B). Indeed, as

$\neg\neg A = A$ under FLP semantics, for normal epistemic specifications the reduct Π^{Φ} amounts to a normal logic program, for which deciding FLP answer set existence is NP-complete. Furthermore, we remark that similar complexity results can be derived for other answer set semantics, and in particular for well-justified FLP answer sets [29].

6. Epistemic well-justified answer set semantics

As shown in [29], FLP answer set semantics of Faber et al. [8] also suffers like other existing answer set semantics such as those defined in [26,32,1,12] from circular justifications for some logic programs without epistemic negation. Consider the following program:

$$\begin{array}{lll} \Pi\colon & p \leftarrow q & r_1 \\ & q \leftarrow \neg q \vee p & r_2 \end{array}$$

Here $I = \{p, q\}$ is an FLP answer set of Π which has a circular justification via the self-supporting loop $p \Leftarrow q \Leftarrow \neg q \vee p \Leftarrow p$, i.e., p being true in I is due to q being true in I (via rule r_1) that is due to I satisfying $\neg q \vee p$ (via rule r_2), which in turn is due to p being true in I.

To the best of our knowledge, the well-justified FLP answer set semantics defined by Shen et al. [29] is the only one in the current literature whose answer sets are *well justified by having a level mapping* for any general logic programs without epistemic negation and thus are free of circular justifications. It is analogous to the standard answer set semantics of Gelfond and Lifschitz [16] for normal logic programs whose answer sets are well justified by having a level mapping [10]. Therefore, in this section we extend the well-justified FLP semantics to an epistemic one and discuss its relation with EFLP semantics.

The well-justified FLP semantics is based on the one-step provability operator $T_{\Pi}(O, N)$, which is an extension of the van Emden–Kowalski one-step provability operator [34] from positive to general logic programs.

Definition 13. (See [29].) Let Π be a logic program without epistemic negation, and let O and N be two first-order theories. Define

$$T_{\Pi}(O, N) = \{head(r) \mid r \in ground(\Pi) \text{ and } O \cup N \models body(r)\}.$$

Informally, $T_{\Pi}(O, N)$ collects all heads of rules in $ground(\Pi)$ whose bodies are entailed by $O \cup N$. When the parameter N is fixed, the entailment relation $\models$ is monotone in O, so $T_{\Pi}(O, N)$ is monotone w.r.t. O, i.e., for any first-order theories $O_1 \subseteq O_2$, we have $T_{\Pi}(O_1, N) \subseteq T_{\Pi}(O_2, N)$. Hence the sequence $\langle T^i_{\Pi}(\emptyset, N)\rangle^{\infty}_{i=0}$, where $T^0_{\Pi}(\emptyset, N) = \emptyset$ and for $i \geq 0$ $T^{i+1}_{\Pi}(\emptyset, N) = T_{\Pi}(T^i_{\Pi}(\emptyset, N), N)$, will converge to a least fixpoint, denoted $lfp(T_{\Pi}(\emptyset, N))$.

The well-justified FLP answer set semantics, abbreviated as *WJ semantics*, is defined in terms of the least fixpoint $lfp(T_{\Pi}(\emptyset, \neg I^-))$ of an interpretation I.

Definition 14. (See [29].) Let I be a model of a logic program Π without epistemic negation. Then I is a *WJ answer set* if $lfp(T_{\Pi}(\emptyset, \neg I^-)) \cup \neg I^- \models A$ for every $A \in I$.

Then, by replacing $\mathcal{X}$ with WJ in Definition 8 we obtain an epistemic well-justified answer set semantics (EWJ semantics for short).

Definition 15 *(EWJ semantics).* Let Π be a logic program and Φ a guess such that Π^{Φ} is a consistent epistemic reduct. The collection $\mathcal{A}$ of all WJ answer sets of Π^{Φ} is a *candidate world view*, resp. a *world view*, of Π w.r.t. Φ if $\mathcal{A}$ satisfies the conditions of Definition 8. Every WJ answer set in a world view is called an *EWJ answer set*.

As an exercise one may check that for the above Examples 2 to 7, their EWJ answer sets are the same as their EFLP answer sets.

Note that a WJ answer set of a logic program without epistemic negation must be an FLP answer set, but the converse is not true; particularly an FLP answer set is a WJ answer set if and only if it is free of circular justifications (see Theorem 6 of Shen et al. [29]). Not surprisingly, such relations between FLP and WJ semantics do not convey to EFLP and EWJ semantics, as illustrated in the following example.

Example 8. Let us first consider the following logic program without epistemic negation:

$$\begin{array}{lll} \Pi\colon & p \leftarrow q \wedge \neg r & r_1 \\ & q \leftarrow \neg q \vee p \vee r & r_2 \\ & r \leftarrow \neg s & r_3 \\ & s \leftarrow \neg r & r_4 \end{array}$$

This program has two FLP answer sets, viz. $I_1 = \{r, q\}$ and $I_2 = \{s, p, q\}$, but only I_1 is a WJ answer set because I_2 has a circular justification via the self-supporting loop $p \Leftarrow q \wedge \neg r \Leftarrow \neg q \vee p \vee r \Leftarrow p$, i.e., p being true in I is due to both q and $\neg r$ being true in I (via r_1), where q being true in I is due to I satisfying $\neg q \vee p \vee r$ (via r_2), which in turn is due to p being true in I.

Next let us add to Π the following rule:

$$t \leftarrow \mathbf{not}\, r \qquad r_5$$

Then this program has a unique world view $\mathcal{A}_1 = \{\{t, s, p, q\}, \{r, q\}\}$ (w.r.t. the guess $\Phi = \{\mathbf{not}\, r\}$) under EFLP semantics, and a unique world view $\mathcal{A}_2 = \{\{r, q\}\}$ (w.r.t. the guess $\Phi = \emptyset$) under EWJ semantics.

As a result, a world view under EWJ semantics is not necessarily a world view under EFLP semantics, and vise versa. However, it is trivial to show that for programs matching normal epistemic specifications, the world views under EWJ and EFLP semantics coincide.

Remark 5. The above example demonstrates that while a semantics $\mathcal{X}$ for logic programs without epistemic negation can be more restrictive than a semantics $\mathcal{Y}$ in the sense that $\mathcal{X}$ always yields a subcollection of the answer sets of $\mathcal{Y}$, this property will not be warranted when these semantics are extended to logic programs with epistemic negation by applying the general framework of Definition 8.

7. Related work

The need for using epistemic negation in knowledge representation and reasoning was long recognized by Gelfond [13,15] and recently further emphasized in [14,33,9,35,19,18,4,37]. In particular, Gelfond [13] first introduced modal operators **K** and **M** to ASP and interpreted them in three-valued interpretations (called *three-valued possible worlds*). Formulas with such modal operators are called *strongly introspective* and logic programs with rules of form (1) called epistemic specifications. Truszczynski [33] revisited this formalism and redefined its semantics in two-valued interpretations.

It turns out that the approaches of Gelfond [13] and Truszczynski [33] suffer from the problem of unintended world views due to recursion through **K**, i.e., they incur epistemic circular justifications for logic programs like $\Pi = \{p \leftarrow \mathbf{K}p\}$, where an unintended world view $\{\{p\}\}$ is produced. To remedy this, Gelfond [14] further updated his program transformation. However, epistemic circular justifications persist for other logic programs such as $\Pi = \{q \leftarrow \neg\mathbf{K}p, p \leftarrow \neg q\}$, where an unintended world view $\{\{p\}\}$ is produced when applying the approaches of Gelfond [14,13] and Truszczynski [33]. In addition, these approaches also suffer from the problem of unintended world views due to recursion through **M** for logic programs like $\Pi = \{p \leftarrow \mathbf{M}p\}$, where an unintended world view $\{\emptyset\}$ is produced.

To address the problems of unintended world views due to recursion through **K** and **M**, Kahl et al. [19,18] proposed a more involved program transformation by appealing to nested expressions of Lifschitz et al. [22]. However this approach has some clear shortcomings as listed in the Introduction. Specifically, as illustrated in Example 1 this approach also suffers from the problem of unintended world views due to recursion through **M**. Moreover, as shown in Shen et al. [29] answer sets of logic programs with nested expressions suffer from circular justifications under the existing semantics as defined in [22,11,12].

In an alternative venue, Wang and Zhang [36] developed *epistemic equilibrium models* for epistemic specifications by introducing modal operators **K** and **M** to Pearce's equilibrium logic [25,26]. This approach suffers from the problems of unintended world views due to recursion through **K** and **M**; for example, both $\{\{p\}\}$ and $\{\emptyset\}$ are world views of $\Pi = \{p \leftarrow \mathbf{K}p\}$ and $\Pi = \{p \leftarrow \mathbf{M}p\}$. Very recently, del Cerro et al. [4] presented a new definition of epistemic equilibrium models (EEMs) and further defined *autoepistemic equilibrium models (AEEMs)* as world views that are maximal EEMs under set inclusion and a special partial preorder over S5 models. A dozen of logic programs were listed all of which except for that program in Example 1 have the AEEMs that coincide with the world views of Kahl [19]; the AEEM of the program in Example 1 coincides with the world view under our EFLP semantics. On the one hand, it is unclear whether AEEMs are free of unintended world views due to recursion through **K** and **M** for general logic programs. On the other hand, as discussed by Shen et al. [29], Pearce's equilibrium semantics coincides with the answer set semantics of Ferraris [11] for propositional logic programs and with that of Ferraris [12] in the first-order case, and these semantics suffer from circular justifications. Since AEEMs are epistemic extensions of Pearce's equilibrium models, circular justifications inevitably convey to AEEMs, thus sometimes leading to undesired world views. For example, for the program $\Pi = \{p \leftarrow \neg\neg p,\ \ p \leftarrow \neg p\}$, $\{p\}$ is a unique minimal equilibrium model/answer set under Pearce [26] and thus $\{\{p\}\}$ is an AEEM under del Cerro et al. [4]. This answer set/world view is undesired because it has a circular justification via the self-supporting loop $p \Leftarrow \neg\neg p \Leftarrow p$. Finally, this approach may miss some desired world views; for instance, $\{\{p\}\}$ is expected to be a world view of the program $\Pi = \{p \leftarrow \mathbf{not}\, p \vee p\}$ (see Example 7), but it is not an AEEM of Π where $\mathbf{not}\, p$ is replaced by $\neg\mathbf{K}p$.

Finally, we mention that Lifschitz [20] defined a modal logic of *Minimal Knowledge and Negation as Failure* (MKNF), which has two modal operators, viz. **K** as defined above and **not** like our epistemic negation operator. MKNF logic has recently been exploited for the integration of description logics and rules in the Semantic Web [24]. As indicated in the end of Lifschitz [20], "MKNF does not cover the important concept of strong introspection introduced in Gelfond [13]." Thus, applying

it to handle epistemic negation would yield unintuitive results. For instance, the logic program Π_1 in Example 2 will be identified with a logic program P_1 consisting of the following formulas:

$$\mathbf{K}\ innocent(John) \vee \mathbf{K}\ guilty(John) \qquad f_1$$
$$\mathbf{not}\ guilty(X) \supset \mathbf{K}\ innocent(X) \qquad f_2$$

Under MKNF, P_1 has two possible collections of models (where each is viewed as an S5 structure):

$$\mathcal{A}_1 = \{\{innocent(John)\}, \{innocent(John), guilty(John)\}\}$$
$$\mathcal{A}_2 = \{\{guilty(John)\}, \{guilty(John), innocent(John)\}\}$$

The first collection means *John* is innocent, while the second says *John* is guilty, which violates our intuition.

8. Summary and future work

We have presented a novel approach to evaluating epistemic negation; specifically, we proposed to apply epistemic negation to minimize the knowledge in world views (knowledge minimization with epistemic negation), introduced a novel program transformation based on epistemic negation, and presented a new definition of epistemic answer set semantics for general logic programs. This approach overcomes both the problem of unintended world views due to recursion through **K** and the problem due to **M**, and thus provides an appropriate solution to epistemic specifications introduced by Gelfond [13] over two decades ago.

Our approach is applicable to extend with epistemic negation any existing answer set semantics such as those defined in [26,27,32,1,8,12,29]. For illustration, we extended FLP answer set semantics of Faber et al. [8] to epistemic FLP semantics. We also extended the well-justified FLP semantics of Shen et al. [29] to an epistemic well-justified semantics.

We showed that for a propositional program Π with epistemic negation, deciding whether Π has EFLP answer sets is Σ_3^p-complete and deciding whether a propositional formula F is true in Π under EFLP semantics is Σ_4^p-complete in general, but has lower complexity for important program classes, such as logic programs that match normal epistemic specifications, where the complexity of world view existence and query evaluation drops by one level in the polynomial hierarchy.

Reasoning with both epistemic and default negations is powerful in expressiveness, but is expensive in computation. Methods for efficient system implementation present a challenging future task. Currently we are considering implementation for normal epistemic specifications; an attractive property of this class of logic programs is that their epistemic reducts amount to a normal logic program under FLP or well-justified FLP semantics, for which deciding answer set existence is NP-complete.

Acknowledgements

We would like to thank the reviewers for their thoughtful and constructive comments to improve this article. This work is supported in part by China National 973 Program 2014CB340301 and NSFC Grant 61379043, and the Austrian Science Fund (FWF) via the projects P24090 and P27730.

Appendix A. Proofs

Proof of Theorem 1. Let $\mathcal{A} = \bigcup_i \mathcal{A}_i$ be the union of all collections $\mathcal{A}_i$ of models of Π; then $\mathcal{A}$ consists of all models of Π. By the condition that for every epistemic negation **not** F in $ground(\Pi)$, F is true in every model of Π, for every interpretation $I \in \mathcal{A}$, as I satisfies all rules in $ground(\Pi)$ w.r.t. some $\mathcal{A}_i$, I also satisfies these rules w.r.t. $\mathcal{A}$. This means $\mathcal{A}$ itself is a collection of models of Π.

For any interpretation I, if $\mathcal{A}_I = \{I\}$ is a collection of models of Π, then $I \in \mathcal{A}$; conversely, if $I \in \mathcal{A}$, i.e., I satisfies all rules in $ground(\Pi)$ w.r.t. $\mathcal{A}$, then I also satisfies these rules w.r.t. $\mathcal{A}_I = \{I\}$ and thus $\mathcal{A}_I = \{I\}$ is a collection of models of Π. Hence for any interpretation I, $\mathcal{A}_I = \{I\}$ is a collection of models of Π iff $I \in \mathcal{A}$ iff I is a model of Π. Moreover, by Proposition 1, for any interpretation I, $\mathcal{A}_I = \{I\}$ is a collection of models of Π iff I is a model of $\Pi^{\neg}$. Thus I is a model of Π iff I is a model of $\Pi^{\neg}$. This concludes the proof. □

Proof of Proposition 2. Consider an EFLP answer set $I \in \mathcal{A}$ w.r.t. guess Φ. As I is an FLP answer set of the epistemic reduct Π^{Φ}, it is a model of Π^{Φ}. Let $\Pi^{\Phi^{\mathbf{not}}}$ be Π^{Φ} with all $\neg F$ replaced by **not** F. Since for every **not** F in $\Pi^{\Phi^{\mathbf{not}}}$, F is true in all $I \in \mathcal{A}$, by Theorem 1, $\Pi^{\Phi^{\mathbf{not}}}$ and Π^{Φ} have the same models. This means I is a model of $\Pi^{\Phi^{\mathbf{not}}}$ w.r.t. $\mathcal{A}$. Note that $\Pi^{\Phi^{\mathbf{not}}}$ is $ground(\Pi)$ with every **not** $F \in \Phi$ replaced by $\top$. Then r is a rule in $ground(\Pi)$ iff $r^{\top}$ is a rule in $\Pi^{\Phi^{\mathbf{not}}}$, where $r^{\top}$ is r with every **not** $F \in \Phi$ replaced by $\top$. Since for every **not** $F \in \Phi$, F is false in some $J \in \mathcal{A}$, I satisfies r w.r.t. $\mathcal{A}$ iff I satisfies $r^{\top}$ w.r.t. $\mathcal{A}$. This shows I is also a model of $ground(\Pi)$ w.r.t. $\mathcal{A}$. Hence $\mathcal{A}$ is a collection of models of Π. □

Proof of Theorem 3. (1) Let $\mathcal{A}$ be a world view of Π w.r.t. a guess Φ. Then $\mathcal{A}$ is also the set of all FLP answer sets of the epistemic reduct Π^{Φ}. Rules in Π^{Φ} are of the form

$$A_0 \leftarrow A_1 \wedge \ldots \wedge A_m \wedge \neg B_1 \wedge \ldots \wedge \neg B_n$$

where every B_i is true in every $I \in \mathcal{A}$ because $\mathbf{not}\, B_i \in Ep(\Pi) \setminus \Phi$. This means Π^{Φ} and Π^{Φ^+} have the same set of FLP answer sets, where Π^{Φ^+} is Π^{Φ} with all rules containing a negative literal $\neg B_i$ removed. Since Π^{Φ^+} is a positive logic program, it has a unique FLP answer set. Hence $\mathcal{A}$ has only one FLP answer set, i.e., $|\mathcal{A}| = 1$.

(2) For normal logic programs, FLP semantics coincides with the standard answer set semantics. So it suffices to show that $\mathcal{A} = \{I\}$ is a world view of Π iff I is a standard answer set of $\Pi^{\neg}$.

We first show that $\mathcal{A} = \{I\}$ is a candidate world view of Π iff I is a standard answer set of $\Pi^{\neg}$.

($\Longrightarrow$) Assume $\mathcal{A} = \{I\}$ is a candidate world view of Π w.r.t. a guess Φ. Then I is also the FLP answer set of the epistemic reduct Π^{Φ}, and for every $\mathbf{not}\, B$ in $ground(\Pi)$, $\mathbf{not}\, B \in \Phi$ iff B is false in I iff $B \notin I$. Let r be a rule in Π^{Φ}. For every negative literal $\neg B_i$ in the rule body of r, B_i is true in I, i.e., $B_i \in I$. Let Π^{Φ^+} be Π^{Φ} with all rules containing a negative literal $\neg B_i$ removed. Then I is also the FLP answer set of Π^{Φ^+}, which is the least model of Π^{Φ^+}. Note that Π^{Φ^+} is in fact the GL-reduct of $\Pi^{\neg}$. This means I is also a standard answer set of $\Pi^{\neg}$.

($\Longleftarrow$) Assume I is a standard answer set of $\Pi^{\neg}$. Let Φ be a guess such that for every $\mathbf{not}\, B$ in $ground(\Pi)$, $\mathbf{not}\, B \in \Phi$ iff B is false in I iff $B \notin I$. Let r be a rule in $ground(\Pi)$ such that I satisfies $body(r)$ w.r.t. $\mathcal{A} = \{I\}$. r must be of the form

$$H \leftarrow A_1 \wedge \ldots \wedge A_m \wedge \mathbf{not}\, B_1 \wedge \ldots \wedge \mathbf{not}\, B_n$$

with every $A_i \in I$ and every $B_i \notin I$ (i.e., $\mathbf{not}\, B_i \in \Phi$). Then $\Pi^{\neg}$ must have a rule r' of the form

$$H \leftarrow A_1 \wedge \ldots \wedge A_m \wedge \neg B_1 \wedge \ldots \wedge \neg B_n.$$

Note that I satisfies $body(r')$. As I is a model of $\Pi^{\neg}$, H is in I. This means I satisfies r w.r.t. $\mathcal{A}$. Hence $\mathcal{A} = \{I\}$ is a candidate world view w.r.t. Φ.

We have proved that $\mathcal{A} = \{I\}$ is a candidate world view of Π iff I is a standard answer set of $\Pi^{\neg}$. To show that $\mathcal{A} = \{I\}$ is a world view of Π iff I is a standard answer set of $\Pi^{\neg}$, it suffices to show that every candidate world view of Π is a world view of Π.

Assume on the contrary that there are two guesses $\Phi' \supset \Phi$ such that $\mathcal{A}' = \{I'\}$ and $\mathcal{A} = \{I\}$ are candidate world views w.r.t. Φ' and Φ, respectively. For each $\mathbf{not}\, B \in \Phi' \setminus \Phi$, $B \notin I'$ and $B \in I$. This means $I' \neq I$. By the above proof, I' and I are standard answer sets of $\Pi^{\neg}$. We have the following two GL-reducts w.r.t. I' and I respectively:

$$(\Pi^{\neg})^{I'} = \{A_0 \leftarrow A_1 \wedge \cdots \wedge A_m \mid A_0 \leftarrow A_1 \wedge \cdots \wedge A_m \wedge \neg B_1 \wedge \cdots \wedge \neg B_n \text{ is a rule in } ground(\Pi^{\neg}) \text{ and for every } B_i, \mathbf{not}\, B_i \text{ is in } \Phi'\},$$

$$(\Pi^{\neg})^{I} = \{A_0 \leftarrow A_1 \wedge \cdots \wedge A_m \mid A_0 \leftarrow A_1 \wedge \cdots \wedge A_m \wedge \neg B_1 \wedge \cdots \wedge \neg B_n \text{ is a rule in } ground(\Pi^{\neg}) \text{ and for every } B_i, \mathbf{not}\, B_i \text{ is in } \Phi\}.$$

As the two GL-reducts are positive logic programs, I' and I are their least fixpoints, respectively. As $\Phi' \supset \Phi$, every rule in $(\Pi^{\neg})^{I}$ must be in $(\Pi^{\neg})^{I'}$. This implies $I \subset I'$ ($I' \neq I$, as shown above), which contradicts the fact that every standard answer set of a normal logic program is a minimal model of the program. We then conclude the proof. □

Proof of Theorem 4. Membership. The proof of D_2^p membership is in the discussion below Theorem 4.

Hardness. The D_2^p-hardness is shown by the reduction of the D_2^p-complete problem where, given a pair (Π_1, Π_2) of programs, we must decide whether Π_1 has some FLP answer set and Π_2 has no FLP answer set. (This result in turn is easily obtained by a reduction of evaluating QBFs of the form $\exists X \forall Y \phi$ to FLP answer set existence of disjunctive logic programs [5].)

Assume w.l.o.g. that Π_1 and Π_2 are on disjoint signatures and let $A, \bar{A}, C$ be fresh atoms. Then we claim that

$$\Pi = \Pi_1 \cup \{H \leftarrow B \wedge \neg A \mid H \leftarrow B \in \Pi_2\} \cup \{A \leftarrow \neg \bar{A},\ \bar{A} \leftarrow \neg A,\ C \leftarrow \mathbf{not}\, A\}$$

has a candidate world view w.r.t. $\Phi = \emptyset$ iff Π_1 has some FLP answer set and Π_2 has no FLP answer set. Notice that the epistemic reduct is

$$\Pi^{\Phi} = (\Pi \setminus \{C \leftarrow \mathbf{not}\, A\}) \cup \{C \leftarrow \neg A\}.$$

($\Longrightarrow$) Suppose Π has a candidate world view $\mathcal{A}$ w.r.t. $\Phi = \emptyset$. Then Π_1 must have an FLP answer set. Furthermore, A must be true in every FLP answer set of Π^{Φ}. In particular, this means that the guess $\bar{A}$ from $A \leftarrow \bar{A}$, $\bar{A} \leftarrow A$, does not lead to an FLP answer set of Π; this means that the program $\{H \leftarrow B \wedge \neg A \mid H \leftarrow B \in \Pi_2\}$ has no FLP answer set, from which in turn it follows that Π_2 has no FLP answer set.

($\Longleftarrow$) For each FLP answer set I_1 of Π_1, the set $I_1 \cup \{A\}$ is an FLP answer set of Π^{Φ}, and if since Π_2 has no FLP answer set, each answer set in the collection $\mathcal{A}$ of FLP answer sets of Π^{Φ} has this form. As Π_1 has some FLP answer set I_1, it follows that $\mathcal{A} \neq \emptyset$ and $\mathbf{not}\, A$ is false in $\mathcal{A}$; hence $\mathcal{A}$ is a candidate world view of Π w.r.t $\Phi = \emptyset$. □

Proof of Theorem 5. (i) Membership. A nondeterministic Turing machine can make a guess $\Phi \subseteq EP(\Pi)$ such that Π has some candidate world view w.r.t. Φ in polynomial time, and then continue to check that Φ is indeed proper with an oracle for D_2^p in polynomial time, by Theorem 4. As each problem in D_2^p can be decided with two calls of an Σ_2^p oracle, it follows that the problem is in Σ_3^p.

Hardness. The Σ_3^p-hardness of deciding candidate world view existence can be shown by a reduction from evaluating QBFs of the form

$$\exists X \forall Y \exists Z \phi, \tag{A.1}$$

where $X = X_1, \ldots, X_{n_X}$, $Y = Y_1, \ldots, Y_{n_Y}$, and $X = Z_1, \ldots, Z_{n_Z}$ are lists (viewed as sets) of distinct atoms, and $\phi = \bigwedge_{j=1}^{k}(L_{j,1} \vee L_{j,2} \vee L_{j,3})$ is a CNF over atoms $X \cup Y \cup Z$, by lifting a reduction in [5] that proved Π_2^p-completeness of deciding whether an atom U is false in all answer sets of a disjunctive logic program. Without loss of generality, we assume that by assigning each $Y_i \in Y$ the value true, the formula ϕ evaluates to true for every assignment to the remaining variables, i.e., the formula $\forall X, Z\phi[Y/\top]$ evaluates to true; hence, regardless of a concrete assignment σ to X the formula $\phi[X/\sigma, Y/\top]$ will be satisfiable.

For each atom $A \in X \cup Y \cup Z$, we introduce a fresh atom $\bar{A}$, and we use further fresh atoms U and V. Let

$$\Pi_X = \{X_i \leftarrow \mathbf{not}\, \bar{X}_i;\ \ \bar{X}_i \leftarrow \mathbf{not}\, X_i \mid X_i \in X\}, \tag{A.2}$$

$$\Pi_Y = \{Y_i \leftarrow \neg \bar{Y}_i;\ \ \bar{Y}_i \leftarrow \neg Y_i \mid Y_i \in Y\}, \tag{A.3}$$

$$\Pi_Z = \{Z_i \vee \bar{Z}_i \mid Z_i \in Z\}, \tag{A.4}$$

$$\Pi_\phi = \{U \leftarrow L_{j,1}^* \wedge L_{j,3}^* \wedge L_{j,3}^* \mid 1 \le j \le k\} \cup \tag{A.5}$$

$$\{Z_i \leftarrow U;\ \ \bar{Z}_i \leftarrow U \mid Z_i \in Z\} \cup \tag{A.6}$$

$$\{V \leftarrow \mathbf{not}\, V, \mathbf{not}\, \neg U\} \tag{A.7}$$

where the operation * converts each positive literal A into $\bar{A}$ and each negative literal $\neg A$ into A. If we suppose that X is void, the program $\Pi_Y \cup \Pi_Z \cup \Pi_\phi \setminus \{(A.7)\}$ amounts to a (variant of) the program Π_Φ in [5] for evaluating $\Psi = \forall Y \exists Z \phi$ such that U is false in all answer sets of Π_Ψ iff Ψ evaluates to true; for each assignment μ to Y, which is guessed by the rules (A.3) the program Π_Ψ has some answer set candidate $I_\mu = \{Y_i \in Y \mid \mu(Y_i) = \mathit{true}\} \cup \{\bar{Y}_i \in Y \mid \mu(Y_i) = \mathit{false}\} \cup Z \cup \bar{Z} \cup \{U\}$, where $\bar{S} = \{\bar{A} \mid A \in S\}$, that is an answer set if $\phi[Y/\sigma]$ is unsatisfiable; otherwise, if $\phi[Y/\mu]$ is satisfiable, some answer set $I \subset I_\mu$ exists and U is false in each such I. By our assumption, for $\mu = \top$ (i.e., $\mu(Y_i) = \mathit{true}$ for every $Y_i \in Y$) such an answer set exists, thus Π_Ψ has some answer set.

Now let us consider the case where X is non-void, and let $\Pi = \Pi_X \cup \Pi_Y \cup \Pi_Z \cup \Pi_\phi$. Intuitively, the rules in Π_X guess an assignment σ to X, and the candidate world view conditions amount to evaluating the QBF $\forall Y \exists Z \phi[X/\sigma]$. Here, the rule (A.7) plays a crucial role; informally, it checks that in all answer sets of $(\Pi \setminus \{(A.7)\})^\Phi$ the atom U is false, without pruning answer sets in which U is true (note that a constraint $\bot \leftarrow \mathbf{not}\, \neg U$ or $V \leftarrow \neg V, \mathbf{not}\, \neg U$ would not work). Indeed, the rule (A.7) can be satisfied only by a guess Φ such that $\Phi \cap \{\mathbf{not}\, V, \mathbf{not}\, \neg U\} = \{\mathbf{not}\, V\}$, which means that $V \leftarrow \neg\neg U$ is in Π^Φ; the candidate world conditions (a)–(c) are then met iff U is false in every answer set of $(\Pi \setminus \{(A.7)\})^\Phi$. Otherwise, if $\mathbf{not}\, V \notin \Phi$, we have a rule $V \leftarrow \neg V, \cdots$ in Π^Φ, and as V occurs in no other rule head, Π^Φ has no answer set in which V is true, and thus Φ violates condition (a) or (c); if $\mathbf{not}\, \neg U, \mathbf{not}\, V \in \Phi$, then V is in Π^Φ, and hence Φ violates condition (b).

Having clarified the working of (A.7) for guesses Φ that have some candidate world views, we can verify that any such guess encodes a truth assignment σ to X. For $\mathbf{not}\, \bar{X}_i$ and $\mathbf{not}\, X_i$, we have four possible cases for $\Phi_j = \Phi \cap \{\mathbf{not}\, \bar{X}_i, \mathbf{not}\, X_i\}$: 1. $\Phi_1 = \emptyset$, 2. $\Phi_2 = \{\mathbf{not}\, \bar{X}_i, \mathbf{not}\, X_i\}$, 3. $\Phi_3 = \{\mathbf{not}\, \bar{X}_i\}$, and 4. $\Phi_4 = \{\mathbf{not}\, X_i\}$. Of these, Φ_1 means that condition (c) is violated, as the program $(\Pi \setminus \{(A.7)\})^\Phi$ will contain rules

$$X_i \leftarrow \neg \bar{X}_i;\ \ \bar{X}_i \leftarrow \neg X_i,$$

i.e., a choice between X_i and $\bar{X}_i$; as these atoms do not occur in other rule heads, our assumption about ϕ implies that $(\Pi \setminus \{(A.7)\})^\Phi$ has answer sets in which X_i resp. $\bar{X}_i$ is false. Likewise, Φ_2 does not yield a candidate world view: Π^Φ contains the facts X_i and $\bar{X}_i$, and thus condition (b) is violated. Thus, only Φ_3 and Φ_4 remain. In case of Φ_3, the reduct Π^Φ contains

$$X_i\,; \quad \bar{X}_i \leftarrow \neg X_i$$

and thus each FLP answer set of Π^Φ must contain X_i but not $\bar{X}_i$; that is, Φ_3 encodes that X_i is true in it. Similarly, for Φ_4 each FLP answer set of Π^Φ must contain $\bar{X}_i$ but not X_i; that is Φ_4 encodes that X_i is false.

Putting things together, if Φ is guess that has some world view, then Φ encodes a truth assignment σ to X such that the formula $\forall Y \exists Z \phi[X/\sigma]$ evaluates to true; that is, $\exists X \forall Y \exists Z \phi$ evaluates to true. Conversely, if σ is a truth assignment to X such that $\forall Y \exists Z \phi[X/\sigma]$ evaluates to true, then the guess $\Phi = \{\mathbf{not}\, \bar{X}_i \mid \sigma(X_i) = \mathit{true}\} \cup \{\mathbf{not}\, X_i \mid \sigma(X_i) = \mathit{false}\} \cup \{\mathbf{not}\, \neg U\}$ has some candidate world view. As Π is clearly constructible in polynomial time from (A.1), this proves Σ_3^p-hardness of deciding candidate world view existence; furthermore, note that the rules in Π match the syntax of (1).

(ii) Follows from (i), as some world view exists iff some candidate world view exists. □

Proof of Theorem 6. (i) Membership. As in the discussion below Theorem 6, it suffices to guess $\Phi \subseteq EP(\Pi)$ such that Π has a candidate world view $\mathcal{A}$ w.r.t. Φ and Π^{Φ} cautiously entails F. From Theorem 4 and [29] it follows that the guess can be verified with an Σ_2^p oracle in polynomial time; this proves membership in Σ_3^p.

Hardness. Σ_3^p-hardness is a trivial from Theorem 5: a given program Π has some candidate world view iff A is true in $\Pi \cup \{A\}$ w.r.t. candidate world views, i.e., in some candidate world view of Π, where A is a fresh atom.

(ii) Membership. For a guess Φ as in (i), we must in addition test that no $\Phi' \supset \Phi$ exists that has a candidate world view; by Theorem 5, the latter is in Π_3^p. Thus in summary, a guess Φ that has a world view $\mathcal{A}$ in which F is true can be verified with a Σ_3^p oracle in polynomial; this proves Σ_4^p membership.

Hardness. Σ_4^p-hardness is shown by generalizing the reduction in Theorem 5 to encode the evaluation of a QBF

$$\exists W \forall X \exists Y \forall Z \psi, \tag{A.8}$$

where $\psi = \bigvee_{j=1}^{k} L_{j,1} \wedge L_{j,2} \wedge L_{j,3}$ is a DNF over atoms $W \cup X \cup Y \cup Z$. Without loss of generality, we assume that ψ is unsatisfiable if we assign each $Y_i \in Y$ the value true, i.e., $\forall W, Y, Z \neg\psi[Y/\top]$ evaluates to true.

Assume for the moment that W is void; then the negation of the QBF $\forall X \exists Y \forall Z \psi$, i.e., $\exists X \forall Y \exists Z \phi$ were $\phi = \neg\psi$, is encoded by the program Π in the proof of Theorem 5. That is, Π has some candidate world view iff $\forall X \exists Y \forall Z \psi$ evaluates to false. A modification of Π yields that maximality testing of a guess Φ is Π_3^p-hard. To this end, let

$$\Pi_0 = \{H \leftarrow B \wedge A \mid H \leftarrow B \in \Pi\} \cup \tag{A.9}$$

$$\{X_i \leftarrow \neg A;\ \ \bar{X}_i \leftarrow \neg A \mid X_i \in X\} \cup \{V \leftarrow \neg A\} \cup \tag{A.10}$$

$$\{A \leftarrow \mathbf{not}\, \neg A\}, \tag{A.11}$$

where A is a fresh atom, and $\Phi = \emptyset$. It is easy to see that Φ has a candidate world view: $I_0 = X \cup \bar{X} \cup \{V\}$ is the single answer set of ${\Pi_0}^{\Phi}$, and for each epistemic negation $\mathbf{not}\, F$ in $EP(\Pi_0) = \{\mathbf{not}\, V, \mathbf{not}\, \neg U, \mathbf{not}\, \neg A, \mathbf{not}\, X_i, \mathbf{not}\, \bar{X}_i \mid X_i \in X\}$ the formula F is true in I.

Furthermore, it holds that no guess $\Phi' \supset \Phi$ has a candidate world view iff Π has no candidate world view. To see the only if part, suppose some $\Phi' \supset \Phi$ has a candidate world view. Then $\mathbf{not}\, \neg A \in \Phi'$ must hold, as otherwise I_0 is the single answer set of $\Pi_0^{\Phi'}$; this would imply $\Phi' = \Phi$, a contradiction. Now $\Pi_0^{\Phi'}$ contains $A \leftarrow$ and amounts to $\Pi^{\Phi'}$ (simply eliminate A); it follows that Π has a candidate world view w.r.t. $\Phi' \cap EP(\Pi)$. Conversely, if Π has a candidate world view w.r.t. Φ, then it is easy to see that Π_0 has a candidate world view w.r.t. $\Phi' = \Phi \cup \{\mathbf{not}\, \neg A\}$.

Thus, Π_0 has a world view w.r.t. $\Phi = \emptyset$ iff the QBF $\forall X \exists Y \forall Z \psi$ evaluates to true (which proves Π_3^p-hardness of world view checking, i.e., deciding given a program Π and $\Phi \subseteq EP(\Pi)$ whether Π has some world view w.r.t. Φ). Note that since any other guess $\Phi' \supset \Phi$ that has a candidate world view w.r.t. Π_0 must contain $\mathbf{not}\, \neg A$, we can equivalently ask whether $\neg A$ is true in Π_0 under EFLP semantics.

Now we generalize Π_0 to accommodate non-void W, i.e., to encode evaluating the QBF (A.8). To this end, we let

$$\Pi_1 = {\Pi_0}^* \cup \{W_i \leftarrow \mathbf{not}\, \bar{W}_i;\ \ \bar{W}_i \leftarrow \mathbf{not}\, W_i \mid W_i \in W\}$$

where $\bar{W}_i$ is a fresh atom for W_i and ${\Pi_0}^*$ is constructed like Π_0 where in the construction of the rules (A.5) of Π each W_i (resp. $\neg W_i$) literal is replaced by $\bar{W}_i$ (resp. W_i).[11] Note that W_i and $\bar{W}_i$ do not occur in Π_0^* in rule heads; combined with the assumption that $\forall W, X, Z \neg\psi[Y/\top]$ evaluates to true, by similar reasoning as in the proof of Theorem 5 for W_i and $\bar{W}_i$ in place of X_i and $\bar{X}_i$, we obtain that every guess $\Phi \subseteq EP(\Pi_1)$ such that Π_1 has a candidate world view w.r.t. Φ must contain exactly one of $\mathbf{not}\, W_i$ and $\mathbf{not}\, \bar{W}_i$, and thus Φ encodes a truth assignment ν to W where $\nu(W_i) = \mathit{true}$ if $\mathbf{not}\, \bar{W}_i \in \Phi$ (as W_i and $\bar{W}_i \leftarrow \neg W_i$ are in Π_1^{Φ}) and $\nu(W_i) = \mathit{false}$ if $\mathbf{not}\, W_i \in \Phi$ (as $\bar{W}_i$ and $W_i \leftarrow \neg \bar{W}_i$ are in Π_1^{Φ}). Furthermore, if $\mathbf{not}\, \neg A \notin \Phi$, then Φ contains no other epistemic negations, and we denote it by Φ_ν; if $\mathbf{not}\, \neg A \in \Phi$, then the QBF $\forall X \exists Y \forall Z \psi[W/\nu]$ must evaluate to false. Thus, if Φ_ν has a world view, then $\forall X \exists Y \forall Z \psi[W/\nu]$ evaluates to true. On the other hand, every truth assignment to W is encoded by some Φ_ν; thus, it follows that some Φ_ν has a world view iff the QBF (A.8) evaluates to true. As $\neg A$ is true in the candidate world view of a guess Φ iff $\Phi = \Phi_\nu$ for some ν, it follows that $\neg A$ is true in Π_1 under EFLP semantics iff the QBF (A.8) evaluates to true. As Π_1 is constructible in polynomial time from (A.8), this proves Σ_4^p-hardness. □

Appendix B. Programs matching normal epistemic specifications

We show that the complexity of world view existence drops by one level in the polynomial hierarchy for *normal epistemic specifications* with rules of the form

$$L_0 \leftarrow L_1 \wedge \ldots \wedge L_m \wedge (\neg)\, \mathbf{not}\, L_{m+1} \wedge \ldots \wedge (\neg)\, \mathbf{not}\, L_n \tag{B.1}$$

[11] As $\phi = \neg\psi$, the final rules (A.9) that emerge from (A.5) are of the form $U \leftarrow L_{j,1}^{\dagger} \wedge L_{j,3}^{\dagger} \wedge L_{j,3}^{\dagger} \wedge A$ where for any atom C we have $C^{\dagger} = C$ and $\neg C^{\dagger} = \bar{C}$.

where L_0 is an atom and each L_i is an atom A or default negated atom $\neg A$; i.e., the rules match the syntax of (1) with $m = 1$ (the modal atom $\mathbf{M}A$ amounts to $\mathbf{not}\,\neg A$, and $\neg\neg A = A$ under FLP semantics).[12]

Formally, we obtain, were $D^p = D_1^p$ is the analog of D_2^p where Σ_2^p/Π_2^p is replaced by NP/coNP:

Theorem 7. *Given a propositional program Π that is a normal epistemic specification and a guess Φ for it, deciding whether Π has a candidate world view w.r.t. Φ is D^p-complete.*

Proof. Membership. The proof of D^p membership is analogous to the one of Theorem 4, where NP/coNP replaces Σ_2^p/Π_2^p.

Hardness. The D^p-hardness follows from the reduction in the proof of Theorem 4: if the pair (Π_1, Π_2) consists of normal logic programs, deciding whether Π_1 has some FLP answer set and Π_2 has no FLP answer set is D^p-complete. Furthermore, the program Π constructed from (Π_1, Π_2) matches a normal epistemic specification. □

Theorem 8. *Given a propositional program Π that is a normal epistemic specification, deciding whether Π has (i) some candidate world view and (ii) some world view are both Σ_2^p-complete.*

Proof. Membership. The membership proof is analogous to the one of Theorem 5, but uses Theorem 8 instead of Theorem 4.

Hardness. The Σ_2^p-hardness is inherited from the reduction in Theorem 5: if we consider a QBF (A.1) in which Z is void, the resulting program Π matches normal epistemic specifications, and has some candidate world view iff $\exists X \forall Y \neg\phi$, where ϕ is a CNF, evaluates to true. Evaluating such QBFs is Σ_2^p-complete, and remains Σ_2^p-hard even if ϕ is unsatisfiable if each $Y_i \in Y$ is assigned true, i.e., the QBF $\forall X \neg\phi[Y/\top]$ evaluates to true. □

Theorem 9. *Given a propositional program Π that is a normal epistemic specification and a propositional formula F, deciding whether F is true in Π (i) w.r.t. candidate world views is Σ_s^p-complete, and (ii) w.r.t. world views, i.e., under EFLP semantics, is Σ_3^p-complete.*

Proof. Membership. The membership proofs are analogous to the one of Theorem 6, but use Theorem 8 instead of Theorem 4 and the fact that for any guess Φ, Π^Φ is a normal logic program, and cautious inference form the answer sets of such programs is coNP-complete [23].

Hardness. The Σ_2^p-hardness of (i) is immediate from the reduction in the proof of item (i) of Theorem 6 and Theorem 8. The Σ_3^p-hardness of (ii) is shown by generalizing the encoding of evaluating a QBF $\exists X \forall Y \psi$ to deciding candidate world existence for a program Π that matches normal epistemic specifications in Theorem 8 in the same way as in the proof of item (ii) in Theorem 6; note that all rules created match normal epistemic specifications. □

References

[1] M. Bartholomew, J. Lee, Y. Meng, First-order extension of the FLP stable model semantics via modified circumscription, in: Proc. 22nd Int'l Joint Conf. Artificial Intelligence, IJCAI-11, AAAI Press/IJCAI, 2011, pp. 724–730.

[2] G. Brewka, T. Eiter, M. Truszczynski, Answer set programming at a glance, Commun. ACM 54 (12) (2011) 92–103.

[3] J. de Bruijn, T. Eiter, H. Tompits, Embedding approaches to combining rules and ontologies into autoepistemic logic, in: Proc. 11th International Conf. Principles of Knowledge Representation and Reasoning, KR 2008, AAAI Press, 2008, pp. 485–495.

[4] L.F. del Cerro, A. Herzig, E.I. Su, Epistemic equilibrium logic, in: Proc. 24th Int'l Joint Conf. Artificial Intelligence, IJCAI-15, AAAI Press/IJCAI, 2015, pp. 2964–2970.

[5] T. Eiter, G. Gottlob, On the computational cost of disjunctive logic programming: propositional case, Ann. Math. Artif. Intell. 15 (3–4) (1995) 289–323.

[6] T. Eiter, G. Ianni, R. Schindlauer, H. Tompits, A uniform integration of higher-order reasoning and external evaluations in answer-set programming, in: IJCAI, 2005, pp. 90–96.

[7] T. Eiter, G. Ianni, T. Lukasiewicz, R. Schindlauer, H. Tompits, Combining answer set programming with description logics for the semantic web, Artif. Intell. 172 (12–13) (2008) 1495–1539.

[8] W. Faber, G. Pfeifer, N. Leone, Semantics and complexity of recursive aggregates in answer set programming, Artif. Intell. 175 (1) (2011) 278–298.

[9] W. Faber, S. Woltran, Manifold answer-set programs and their applications, in: M. Balduccini, T.C. Son (Eds.), Logic Programming, Knowledge Representation, and Nonmonotonic Reasoning, in: Lect. Notes Comput. Sci., vol. 6565, Springer, 2011, pp. 44–63.

[10] F. Fages, Consistency of Clark's completion and existence of stable models, J. Methods Log. Comput. Sci. 1 (1994) 51–60.

[11] P. Ferraris, Answer sets for propositional theories, in: Proc. 8th International Conf. on Logic Programming and Nonmonotonic Reasoning, LPNMR 2005, in: Lect. Notes Comput. Sci., vol. 3662, Springer, 2005, pp. 119–131.

[12] P. Ferraris, J. Lee, V. Lifschitz, Stable models and circumscription, Artif. Intell. 175 (1) (2011) 236–263.

[13] M. Gelfond, Strong introspection, in: Proc. 9th National Conf. Artificial Intelligence, AAAI 1991, AAAI Press/The MIT Press, 1991, pp. 386–391.

[14] M. Gelfond, New semantics for epistemic specifications, in: Proc. 11th International Conf. Logic Programming and Nonmonotonic Reasoning, LPNMR 2011, in: Lect. Notes Comput. Sci., vol. 6645, 2011, pp. 260–265.

[15] M. Gelfond, Logic programming and reasoning with incomplete information, Ann. Math. Artif. Intell. 12 (1–2) (1994) 89–116.

[16] M. Gelfond, V. Lifschitz, The stable model semantics for logic programming, in: Proc. 5th International Joint Conference and Symposium on Logic Programming, IJCSLP 1988, MIT Press, 1988, pp. 1070–1080.

[17] M. Gelfond, V. Lifschitz, Classical negation in logic programs and disjunctive databases, New Gener. Comput. 9 (1991) 365–385.

[12] Recall that we omit strong negation $\sim$; however, strongly negated literals $\sim A$ can be compiled away in polynomial time (cf. Example 3), and thus the results of this section extend to a respective extension.

[18] P. Kahl, R. Watson, E. Balai, M. Gelfond, Y. Zhang, The language of epistemic specifications (refined) including a prototype solver, J. Log. Comput. (2015), http://dx.doi.org/10.1093/logcom/exv065.
[19] P.T. Kahl, Refining the semantics for epistemic logic programs, Ph.D. thesis, Texas Tech University, USA, 2014.
[20] V. Lifschitz, Nonmonotonic databases and epistemic queries, in: Proc. 12th International Joint Conf. Artificial Intelligence, IJCAI-91, Morgan Kaufmann, 1991, pp. 381–386.
[21] V. Lifschitz, Thirteen definitions of a stable model, in: Fields of Logic and Computation, in: Lect. Notes Comput. Sci., vol. 6300, Springer, 2010, pp. 488–503.
[22] V. Lifschitz, L.R. Tang, H. Turner, Nested expressions in logic programs, Ann. Math. Artif. Intell. 25 (1–2) (1999) 369–389.
[23] W. Marek, M. Truszczyński, Autoepistemic logic, J. ACM 38 (3) (1991) 588–619.
[24] B. Motik, R. Rosati, Reconciling description logics and rules, J. ACM 57 (5) (2010) 1–62.
[25] D. Pearce, A new logical characterisation of stable models and answer sets, in: Non-Monotonic Extensions of Logic Programming, Selected Papers, NMELP'96, in: Lect. Notes Comput. Sci., vol. 1216, Springer, 1996, pp. 57–70.
[26] D. Pearce, Equilibrium logic, Ann. Math. Artif. Intell. 47 (1–2) (2006) 3–41.
[27] W. Pelov, M. Denecker, M. Bruynooghe, Well-founded and stable semantics of logic programs with aggregates, Theory Pract. Log. Program. 7 (3) (2007) 301–353.
[28] R. Reiter, On closed world data bases, in: H. Gallaire, J. Minker (Eds.), Logic and Data Bases, Plenum, New York, 1978, pp. 119–140.
[29] Y.D. Shen, K. Wang, T. Eiter, M. Fink, C. Redl, T. Krennwallner, J. Deng, FLP answer set semantics without circular justifications for general logic programs, Artif. Intell. 213 (2014) 1–41.
[30] T.C. Son, E. Pontelli, P.H. Tu, Answer sets for logic programs with arbitrary abstract constraint atoms, J. Artif. Intell. Res. 29 (2007) 353–389.
[31] E. Sperner, Ein Satz über Untermengen einer endlichen Menge, Math. Z. XXVII (1928) 544–548.
[32] M. Truszczynski, Reducts of propositional theories, satisfiability relations, and generalizations of semantics of logic programs, Artif. Intell. 174 (16–17) (2010) 1285–1306.
[33] M. Truszczynski, Revisiting epistemic specifications, in: Logic Programming, Knowledge Representation, and Nonmonotonic Reasoning, 2011, pp. 315–333.
[34] M.H. van Emden, R.A. Kowalski, The semantics of predicate logic as a programming language, J. ACM 23 (4) (1976) 733–742.
[35] H. Vlaeminck, J. Vennekens, M. Bruynooghe, M. Denecker, Ordered epistemic logic: semantics, complexity and applications, in: Proc. 13th International Conf. Principles of Knowledge Representation and Reasoning, KR 2012, AAAI Press, 2012, pp. 369–379.
[36] K. Wang, Y. Zhang, Nested epistemic logic programs, in: Proc. 8th International Conf. on Logic Programming and Nonmonotonic Reasoning, LPNMR 2005, in: Lect. Notes Comput. Sci., vol. 3662, Springer, 2005, pp. 279–290.
[37] Z. Zhang, S. Zhang, Logic Programming with Graded Modality, in: Proc. 13th International Conference on Logic Programming and Nonmonotonic Reasoning, LPNMR, 2015, pp. 27–30.