# Epistemic Answer Set Programming

Ezgi Iraz Su

# Epistemic Answer Set Programming

Ezgi Iraz Su[1,2(✉)]

[1] Department of Computer Engineering, Istinye University, Istanbul, Turkey
Ezgi-Iraz.Su@irit.fr
[2] IRIT, University of Toulouse, Toulouse, France

**Abstract.** This paper introduces a new epistemic extension of answer set programming (ASP) called *epistemic ASP* (E-ASP). Then, it compares E-ASP with existing approaches, showing the advantages and the novelties of the new semantics and discusses which formalisms provide more intuitive results: compared to Gelfond's epistemic specifications (ES), E-ASP defines a simpler, but sufficiently strong language. Its *epistemic view* semantics is a natural and more standard generalisation of ASP's original answer set semantics, so it allows for ASP's previous language extensions. Moreover, compared to all semantics proposals in the literature, epistemic view semantics facilitates understanding of the intuitive meaning of epistemic logic programs and solves unintended results discussed in the literature, especially for epistemic logic programs including constraints.

**Keywords:** Answer set programming · Epistemic specifications · Modal logic S5 · Stable models · Answer sets · World views · Autoepistemic equilibrium models

## 1 Introduction

Logic programming (LP) [21] unifies different areas of computation by exploiting the greater generality of logic. Answer set programming (ASP) [13,14,24,25] is an approach to declarative programming, and it relates LP to declarative problem solving by *answer sets*—consistent sets $A$ of literals[1] in which $p \notin A$ or $\sim p \notin A$. In a sense, they are *partial* valuations. Some researchers prefer to call them *3-valued*. For instance, empty valuation assigns neither *true* nor *false* to a propositional variable $p$, leaving it *undetermined*, which is characterised by *negation as failure* (NAF) [6,7] in ASP. Answer set semantics [10] of ASP has

I want to thank Andreas Herzig, Luis Fariñas del Cerro, Michael Gelfond, Patrick Thor Kahl, Thomas Eiter, Yi-Dong Shen, Pedro Cabalar, and Jorge Fandinno for their research related to this paper and the anonymous reviewers for their valued comments on the drafts of this work.

[1] In ASP, a literal is a propositional variable $p$ or a *strongly-negated* propositional variable $\sim p$.

https://doi.org/10.1007/978-3-030-19570-0_40

provided a correct interpretation of NAF and related ASP to nonmonotonic reasoning [28,32]. ASP is currently central to various approaches in nonmonotonic reasoning with a wide range of applications in science and technology.

Despite its successes, ASP has some drawbacks. Among others, ASP is not powerful enough to correctly represent incomplete information, exactly in situations where there are multiple answer sets of an ASP program because NAF performs locally (separately in each answer set) and cannot reason about over a whole range of answer sets. Since 1991, a considerable amount of ASP research has focused on the problem of incomplete information [3,5,8,9,11,17–20,22,29–31,33–39], none of which however resulted in a fully satisfactory semantics. We attack the same problem in order to overcome the obstacles of the previous approaches towards a solution.

The first approach of this line of research is Gelfond's *epistemic specifications* (ES) [8]: he extended a special form of ASP called disjunctive logic programming [14] by epistemic operators, able to quantify over *belief set* (which is, in structure, analogous to an answer set) collections. The interpretation of this new language is in terms of a *world view*—a maximal collection of belief sets about a world reflected by an epistemic logic program $\Pi$. Hence, a world view is a kind of 3-valued S5 model (set of valuations). Similar to the answer set semantics, the world view semantics is also reduct-based. However, different from the reduct definition of the former where we only eliminate NAF, here the goal is, in principle, to remove epistemic modal operators. Thus, the reduct $\Pi^{\mathcal{A}}$ of an epistemic logic program $\Pi$ w.r.t. a world view candidate $\mathcal{A}$ is an ASP program which may include NAF as well. In both semantics, the selection of these special models from among all models of a program is in two steps (plus a fixed point check): first, we compute the reduct of a program by a candidate model (a valuation or a set of valuations, depending on the context); second, we construct the collection $\mathcal{A}$ of all answer sets of this reduct. If the candidate model which is a (possibly empty) set of literals in ASP, is an element of $\mathcal{A}$, then we call it an *answer set*. In ES, if the candidate model (which is similar to $\mathcal{A}$ in structure) equals $\mathcal{A}$, then we call it a *world view* of the original program. Thus, the ultimate decision follows a sort of fixed point construction.

Since Gelfond's first version [8], several semantics proposals have been suggested for ES. The majority are reduct-based world view semantics: while some offer a slightly different refinement of the preceding approach [9,11,17–20] in order to correct unintended results, some others propose significantly different definitions of reducts and world views [29,30]. There is also another kind of approach, inspired by the Kripke semantics of modal logics over a more general language [5,34]. The rest [4,33,35] are based on an epistemic extension of equilibrium model approach [26,27]. As [35] embeds Gelfond's obsolete previous version [9], it is out of our consideration. [4] contains a refinement of [33], and [4] is somewhat successful in providing intuitive results. To sum up, the (to a certain extent) successful approaches of the day are [4,20,29].

In this paper, we propose a novel epistemic extension of ASP called *epistemic ASP* (E-ASP), which has a modest, but neat syntax character, compared

to Gelfond's ES. However, our language is expressive enough to formulate all motivating examples of ES. The semantics of this new language is given by *epistemic views*, which are, in structure, similar to world views. The main advantage of our approach over previous semantics is its simplicity and similarity to answer set semantics of ASP. Moreover, it performs well both with cyclic and acyclic programs, giving intuitive results. Especially, it offers a solution to the recent constraint problem discussed in the literature [3,19].

The paper is organised as follows. The first 3 sections present related work: Sect. 2 recalls ES and its world view semantics. Section 3 introduces autoepistemic equilibrium models as an alternative to world views. Section 4 defines epistemic negation, with which suggests a new reduct and world view definition. Section 5 includes the main contribution of this paper, where we introduce E-ASP. We propose epistemic view semantics and compare our results with those of [4,19,29]. Section 6 recalls epistemic splitting property by Cabalar et al. Section 7 concludes the paper with future work plan.

## 2 Epistemic Specifications (ES) and Its World View Semantics

We here recall the recent versions of Gelfond's ES, suggested by Kahl et al. [18–20].

### 2.1 The Language of ES ($\mathcal{L}_{\mathsf{ES}}$)

The language $\mathcal{L}_{\mathsf{ES}}$ extends that of ASP [1] by the modalities K ('known') and M ('may be true'). Literals of $\mathcal{L}_{\mathsf{ES}}$ are divided into four kinds: *objective literals* ($l$), *extended objective literals* ($L$), *subjective literals* ($g$) and *extended subjective literals* ($G$).

| $l$ | $L$ | $g$ | $G$ |
|---|---|---|---|
| $p \mid {\sim}p$ | $l \mid \texttt{not}\, l$ | $\mathsf{K}\, l \mid \mathsf{M}\, l$ | $g \mid \texttt{not}\, g$ |

where $p$ ranges over a set $\mathbb{P}$ of propositional variables[2]. $\mathcal{L}_{\mathsf{ES}}$ has two negations; strong negation $\sim$ and NAF (aka, default negation) $\texttt{not}$: $\texttt{not}\, \varphi$ is read "$\varphi$ is false by default".

A *rule* is a logical statement of the form '*head* $\leftarrow$ *body*'. In particular, a rule $\rho$ of ES has the structure '$l_1 \,\texttt{or}\, \ldots \,\texttt{or}\, l_m \leftarrow e_1, \ldots, e_n$' in which $body(\rho)$ viz. '$e_1, \ldots, e_n$' is made up of arbitrary ES literals whereas $head(\rho)$ viz. '$l_1 \,\texttt{or}\, \ldots \,\texttt{or}\, l_m$' is composed of only objective literals. When $m = 0$, we suppose $head(\rho)$ to be $\bot$ and call the rule $\rho$ a *constraint*. When $n = 0$, we suppose $body(\rho)$ to be $\top$ and call the rule $\rho$ a *fact*.

[2] The use of variables in ES is understood as abbreviations for the collection of their ground instances. Thus, for simplicity, we restrict here the language $\mathcal{L}_{\mathsf{ES}}$ to the propositional case.

An *epistemic logic program* is a finite collection of the rules of ES. Here is Gelfond's *eligibility* program $\Pi_G$, motivating us for the need of modal operators in ASP[3]:

$$\Pi_G = \Big\{ e \leftarrow h \,|\, e \leftarrow f, m \,|\, {\sim}e \leftarrow {\sim}h, {\sim}f \,|\, h \,\mathtt{or}\, f \leftarrow \;|\, i \leftarrow \mathtt{not}\,\mathsf{K}\, e, \mathtt{not}\,\mathsf{K}\,{\sim}e \quad (1)$$

in which $h$ stands for *highGPA*; $f$ for *fairGPA*; $e$ for *eligible*; $m$ for *minority*; $i$ for *interview*. The first three rules of $\Pi_G$ indicate the college rules to decide eligibility for scholarship. When we consider these rules with a database, consisting of a disjunctive information given by the fact '$h \,\mathtt{or}\, f$' for a specific student, note that NAF alone is not sufficient anymore to formalise the statement "*a student whose eligibility is not determined by the college rules should be interviewed*". The correct representation is given by using modalities, able to quantify over belief sets, viz. '$i \leftarrow \mathtt{not}\,\mathsf{K}\, e, \mathtt{not}\,\mathsf{K}\,{\sim}e$'.

### 2.2 The Semantics of ES

Let $\mathcal{A}$ be a non-empty collection of consistent sets of objective literals, and let $A \in \mathcal{A}$. Satisfaction of literals is defined by: for an objective literal $l$ and a subjective literal $g$,

$\mathcal{A}, A \models_{\mathsf{ES}} l$ if $l \in A$;
$\mathcal{A}, A \models_{\mathsf{ES}} \mathsf{K}\, l$ if $l \in A$ for every $A \in \mathcal{A}$;
$\mathcal{A}, A \models_{\mathsf{ES}} \mathsf{M}\, l$ if $l \in A$ for some $A \in \mathcal{A}$;

$\mathcal{A}, A \models_{\mathsf{ES}} \mathtt{not}\, l$ if $l \notin A$.
$\mathcal{A}, A \models_{\mathsf{ES}} \mathtt{not}\, g$ if $\mathcal{A} \not\models_{\mathsf{ES}} g$.

Note that satisfaction of an objective literal $l$ is independent of $\mathcal{A}$, while satisfaction of a subjective literal $g$ is independent of $A$. Thus, we sometimes write $\mathcal{A} \models_{\mathsf{ES}} g$ or $A \models_{\mathsf{ES}} l$. Then, satisfaction of an epistemic logic program $\Pi$ is defined by: for every rule $\rho \in \Pi$,

$$\mathcal{A}, A \models_{\mathsf{ES}} \rho \quad \text{viz.} \quad \text{"}\mathcal{A}, A \models_{\mathsf{ES}} body(\rho) \quad \text{implies} \quad \mathcal{A}, A \models_{\mathsf{ES}} head(\rho)\text{"}.$$

Finally, given an epistemic logic program $\Pi$, whether $\mathcal{A}$ is a world view of $\Pi$ is decided as follows: we first compute the reduct $\Pi^{\mathcal{A}} = \{\rho^{\mathcal{A}} \,:\, \rho \in \Pi\}$ of $\Pi$ with respect to $\mathcal{A}$, in which we eliminate K and M according to Table 1. Then, $\mathcal{A}$ is a *world view* of $\Pi$ (w.r.t. [18]) if $\mathcal{A} = \mathtt{AS}(\Pi^{\mathcal{A}})$ where $\mathtt{AS}(\Pi^{\mathcal{A}})$ denotes the set of all answer sets of $\Pi^{\mathcal{A}}$.

For example, the only world view of $\Pi_G$ (see (1)) is $\big\{\{h, e, i\}, \{f, i\}\big\}$. Table 2 contains more examples with focus on disjunctive information, among which the ones emphasising the necessity for a refinement of Gelfond's versions [8,9,11] are given in bold.

Then, Kahl extended his version [18] to allow for expressions, formed from objective literals being preceded by a sequence of not's and a modal operator (K or M). Major forms of such expressions are included in Tables 3, 4 and 5, together with the equivalence relations they are involved in.

[3] We use '|' (a bit informally) to separate the rules of a program in this paper.

**Table 1.** Kahl's definition of reduct

| literal $G$ | if $\mathcal{A} \models_{\mathsf{ES}} G$ | if $\mathcal{A} \not\models_{\mathsf{ES}} G$ |
|---|---|---|
| $\mathsf{K}\, l$ | replace by $l$ | replace by $\bot$ |
| $\mathsf{M}\, l$ | replace by $\top$ | replace by $\texttt{not not}\ l$ |
| $\texttt{not}\ \mathsf{K}\, l$ | replace by $\top$ | replace by $\texttt{not}\ l$ |
| $\texttt{not}\ \mathsf{M}\, l$ | replace by $\texttt{not}\ l$ | replace by $\bot$ |

**Table 2.** Epistemic logic programs and their world views

| epistemic logic program $\Pi$ | world views |
|---|---|
| $p\ \texttt{or}\ q \leftarrow$ | $\{\{p\},\{q\}\}$ |
| $p \leftarrow \texttt{not}\ \mathsf{K}\, p$ | no world view |
| $\boldsymbol{p \leftarrow \mathsf{K}\, p}$ | $\{\emptyset\}$ (in [8,9] : $\{\emptyset\}$ and $\{\{p\}\}$) |
| $\boldsymbol{p \leftarrow \mathsf{M}\, p}$ | $\{\{p\}\}$ (in [8,9,11] : $\{\emptyset\}$ and $\{\{p\}\}$) |
| $\boldsymbol{q \leftarrow\ \texttt{not}\ Kp\ \mid\ p \leftarrow\ \texttt{not}\ q}$ | $\{\{q\}\}$ (in [8,9,11] : $\{\{p\}\}$ and $\{\{q\}\}$) |
| $p \leftarrow \texttt{not}\ Kq\ \mid\ q \leftarrow \texttt{not}\ Kp$ | $\{\{p\}\}$ and $\{\{q\}\}$ |
| $\boldsymbol{p\ \texttt{or}\ q \leftarrow\ \mid\ p \leftarrow \mathsf{M}\, q}$ | $\{\{p\}\}$ (in [8,9,11] : no world view) |
| $p\ \texttt{or}\ q \leftarrow\ \mid\ p \leftarrow \texttt{not}\ \mathsf{K}\, q$ | $\{\{p\}\}$ |
| $p\ \texttt{or}\ q \leftarrow\ \mid\ r \leftarrow \mathsf{K}\, p$ | $\{\{p\},\{q\}\}$ |
| $p\ \texttt{or}\ q \leftarrow\ \mid\ p \leftarrow \mathsf{K}\, q$ | $\{\{p\},\{q\}\}$ |
| $p\ \texttt{or}\ q \leftarrow\ \mid\ p \leftarrow \texttt{not}\ \mathsf{M}\, q$ | $\{\{p\},\{q\}\}$ |

**Table 3.** Equivalence relations of multiply negated literals by NAF

| $exp_1$ | $exp_2$ | Comment |
|---|---|---|
| $\texttt{not}\ l$ | $\texttt{not not not}\ l$ | an odd number of $\texttt{not} \equiv \texttt{not}$ |
| $\texttt{not not}\ l$ | $\texttt{not not not not}\ l$ | an even number ($> 0$) of $\texttt{not} \equiv \texttt{not not}$ |

**Table 4.** Equivalence relations of extended subjective literals

| $exp_1$ | $exp_2$ | Comment |
|---|---|---|
| $\mathsf{K}\, l$ | $\texttt{not}\ \mathsf{M}\ \texttt{not}\ l$ | $\mathsf{K}$ and $\mathsf{M}$ are dual: they can be expressed interchangeably. |
| $\mathsf{M}\, l$ | $\texttt{not}\ \mathsf{K}\ \texttt{not}\ l$ | |
| $\texttt{not}\ \mathsf{K}\, l$ | $\mathsf{M}\ \texttt{not}\ l$ | |
| $\texttt{not}\ \mathsf{M}\, l$ | $\mathsf{K}\ \texttt{not}\ l$ | |

**Table 5.** Equivalences with modal operators and double `not`

| $exp_1$ | $exp_2$ | $exp_3$ | Comment |
|---|---|---|---|
| $\mathsf{K}\, l$ | $\mathsf{K}\ \texttt{not not}\ l$ | $\texttt{not not}\ \mathsf{K}\, l$ | NAF behaves classically here: double '`not`' vanishes. |
| $\mathsf{M}\, l$ | $\mathsf{M}\ \texttt{not not}\ l$ | $\texttt{not not}\ \mathsf{M}\, l$ | |

Soon after, [4] pointed at another program, $\Pi = \{p \leftarrow \mathsf{M}\, q, \mathtt{not}\, q \mid q \leftarrow \mathsf{M}\, p, \mathtt{not}\, p$ giving unintended world views under Kahl's refined version. Indeed, [18] proposes two world views $\{\emptyset\}$ and $\{\{p\}, \{q\}\}$, of which the former seems to be unintended. Following this example, Kahl et al. [20] came up with another update to address the issue, with semantics supporting only the latter: inspired by [29] (see Sect. 4), they first define

$$\mathtt{Ep}(\Pi) = \{G \ : \ G = \mathtt{not}\, \mathsf{K}\, L \text{ for some extended objective literal } L \text{ and } G \text{ appears in } \Pi\}.$$

Note that the set $\mathtt{Ep}(\Pi)$ checks all extended subjective literals occurring in $\Pi$ and by using the equivalences between $\mathtt{notnot}\mathsf{K}\, p$ and $\mathsf{K}\, p$, as well as $\mathtt{notK}\mathtt{not} p$ and $\mathsf{M}\, p$, picks the forms $\mathtt{not}\, \mathsf{K}\, L$ (for an extended objective literal $L$) in their structure. To illustrate this set, consider the program $\Gamma = \{t \leftarrow \mathsf{K}\, p, \mathsf{M}\, q, \mathtt{not}\mathsf{K}\, r, \mathtt{not}\mathsf{M}\, s\}$. Thus, $\mathtt{Ep}(\Gamma) = \{\mathtt{not}\mathsf{K}\, p, \mathtt{not}\mathsf{K}\mathtt{not} q, \mathtt{not}\mathsf{K}\, r, \mathtt{not}\mathsf{K}\mathtt{not} s\}$. Then, they take the subset $\Phi_{\mathcal{A}} = \{G \in \mathtt{Ep}(\Pi) \ : \ \mathcal{A} \models_{\mathsf{ES}} G\}$ w.r.t. a candidate model $\mathcal{A}$. Finally, $\mathcal{A}$ is a world view of $\Pi$ if:

$$\mathcal{A} = \mathtt{AS}(\Pi^{\mathcal{A}}), \text{ and there is no } \mathcal{A}' \text{ such that } \mathcal{A}' = \mathtt{AS}(\Pi^{\mathcal{A}'}) \text{ and } \Phi_{\mathcal{A}'} \supset \Phi_{\mathcal{A}}.$$

On the one hand, as mentioned in [3,19], researchers have now discovered another problem: different from their effect on answer sets in ASP[4], in ES inserting a constraint into a program may now bring out completely new world views. The reason is because here constraints show their effect on its belief sets rather than a world view as a whole. So, not only Kahl's all versions, but also [4,29] suffer from new counterintuitive results produced over acyclic programs while they are trying to obtain the intuitive understanding of the behaviour of cycles. Interestingly, only in Gelfond's first version [8,9], and its generalisation [34] by Truszczyński, constraints function to rule out world views, violating that constraint (as desired). To sum up, as a negative outcome of added complexity, the recent semantics approaches seem to have lost this property. On the other hand, as argued in [29], Kahl's reduct definition offers a complex program transformation, lacking an intuitive explanation for the replacement of subjective literals.

Let us terminate our discussion with two examples: $\Pi_1 = \{p \leftarrow \mathsf{K}\, p \mid p \leftarrow \mathtt{not}\, \mathsf{K}\, p$ and $\Pi_2 = \{p \leftarrow \mathsf{M}\, p \mid p \leftarrow \mathtt{not}\, \mathsf{M}\, p$. As Kahl obtains no world view for $\Pi_1$, he gets a unique world view $\{\{p\}\}$ for $\Pi_2$. However, the body parts of these programs produce a tautology (see the equivalence relations given in Table 5), so it is strange to see two different solutions according to his semantics approach, but not $\{\{p\}\}$ only for both.

[4] In ASP, constraints show their effect on programs by eliminating or keeping their answer sets.

# 3 Fariñas et al.'s Approach: Autoepistemic Equilibrium Models

In 2015, Fariñas et al. [4] proposed the *autoepistemic equilibrium models* (AEEMs) approach as an alternative semantics for ES. This section briefly recalls their approach.

## 3.1 Epistemic Here-and-There Logic (EHT) and Its Equilibrium Models

EHT extends the logic of here-and-there (HT) [15] by (nondual) epistemic modal operators K and $\hat{\mathsf{K}}$, of which K is the same as K in ES, but [4] never explains the meaning and reading of $\hat{\mathsf{K}}$. Note that as shown later via an example, the modal operator M in ES is translated to $\neg\mathsf{K}\neg$ in EHT. An EHT model is a collection of HT models. It can also be described as a refinement of S5 models (sets of valuations) [2] in which valuations are replaced by HT models. Formally, an *EHT model* is an ordered pair $\langle \mathcal{T}, \hbar \rangle$ in which

- $\mathcal{T} \subseteq 2^{\mathbb{P}}$ is a nonempty set of valuations (i.e., a classical S5 model);
- $\hbar : \mathcal{T} \to 2^{\mathbb{P}}$ is a map, assigning to each *there*-world $T \in \mathcal{T}$ a *here*-world $\hbar(T) \subseteq T$.

*Epistemic equilibrium models* (EEMs) of a formula $\varphi \in \mathcal{L}_{\mathsf{EHT}}$ are then defined as particular S5 models satisfying a minimality condition [4] (similar to that of [27])[5]:

$$\mathtt{EEM}(\varphi) = \big\{\mathcal{T} \subseteq 2^{\mathbb{P}} \ : \ \mathcal{T}, \mathcal{T} \models_{\mathsf{S5}} \varphi \text{ and there is no } h \neq id \text{ such that } \langle \mathcal{T}, \hbar \rangle, \mathcal{T} \models_{\mathsf{EHT}} \varphi\big\}.$$

A typical ES program $\Pi$ is translated into an EHT theory $\Pi^*$ via a map $(.)^*$ as in:

$$\Pi = \Big\{ p \,\mathtt{or}\, {\sim}q \leftarrow \mathsf{M}\, r, \mathtt{not}\, s \mid q \leftarrow \mathtt{not}\, \mathsf{K}\, p$$
$$\Pi^* = \big((\neg\mathsf{K}\neg r \wedge \neg s) \rightarrow (p \vee \widetilde{q})\big) \ \wedge \ \big(\neg\mathsf{K}\, p \rightarrow q\big) \ \wedge \ \neg\big(q \wedge \widetilde{q}\big).$$

The EEM approach fails to give intuitive results, especially in the presence of disjunction. Gelfond's example $\Pi_G$ (see (1)) immediately supports this fact (see Table 6 for more examples): $\Pi_G$ has a unique world view $\big\{\{h, e, i\}, \{f, i\}\big\}$, but $\Pi_G^*$ has three EEMs: $\mathcal{T}_1 = \big\{\{h, e, i\}, \{f, i\}\big\}$, $\mathcal{T}_2 = \big\{\{h, e\}\big\}$ and $\mathcal{T}_3 = \big\{\{f, i\}\big\}$, among which $\mathcal{T}_2$ is unintended. To overcome this problem, [4] uses a *selection process* over EEMs and proposes *autoepistemic equilibrium models* (AEEMs)[6].

[5] For the truth conditions of EHT, you can refer to [4].

[6] However, as in Kahl's approach, adding a constraint into a program may also give here unexpected results. For instance, take the eligibility program $\Pi_G$ and a constraint $\leftarrow i$. Then, the resulting EHT theory $\Pi_G^* \cup \{\neg i\}$ has a unique AEEM $\mathcal{T}_2 = \big\{\{h, e\}\big\}$, instead of having no AEEM.

The AEEM approach can handle a more general language, but its way of choosing intuitive models is highly complex: the AEEM semantics depends on two orderings, set inclusion $\subseteq$ and a preference ordering $\leq_\varphi$, which function simultaneously. So, it is possible in principle that two EEMs can eliminate each other w.r.t. their different orderings. To spell it out, it could happen that $\mathcal{T}_1 \subset \mathcal{T}_2$, but also $\mathcal{T}_2 <_\varphi \mathcal{T}_1$ for $\mathcal{T}_1, \mathcal{T}_2 \in \mathtt{EEM}(\varphi)$. For instance, let $\varphi = p \vee r \vee \mathsf{K}\,(p \vee q)$. Then, $\mathtt{EEM}(\varphi) = \big\{\{\{p\}\}, \{\{r\}\}, \{\{q\}\}, \{\{p\}, \{q\}\}, \{\{p\}, \{r\}\}\big\}$, among which there are two satisfying the condition above: $\big\{\{p\}, \{q\}\big\} <_\varphi \big\{\{p\}\big\}$ and $\big\{\{p\}\big\} \subset \big\{\{p\}, \{q\}\big\}$. Fortunately, in this case, $\mathtt{AEEM}(\varphi) = \big\{\{\{p\}, \{r\}\}\big\}$ since $\big\{\{p\}\big\} \subset \big\{\{p\}, \{r\}\big\}$ and $\big\{\{p\}, \{q\}\big\} <_\varphi \big\{\{p\}, \{r\}\big\}$. One immediate question is if $\big\{\{p\}, \{q\}\big\}$ is indeed unintended. Briefly, this approach may be suffering from such a clash in the selection process although none has been found so far.

## 4 Shen and Eiter's Approach: Epistemic Negation

In 2016, Shen et al. [29,30] proposed a new semantics for ES. The idea is to use notK (which they call *epistemic negation*) to minimise knowledge in the set of all belief sets. Given an epistemic logic program $\Pi$ and a nonempty collection $\mathcal{A} \subseteq 2^{\mathbb{P}}$ of consistent sets of objective literals, let $\mathtt{Ep}(\Pi)$ (see Sect. 2.2) be the set of all epistemic negations appearing in $\Pi$, and let $\Phi \subseteq \mathtt{Ep}(\Pi)$ be its subset (which they call a *guess*). Let $\Phi_{\mathcal{A}} = \{G \in \mathtt{Ep}(\Pi) \ : \ \mathcal{A} \models G\}$ be the set of all epistemic negations in $\Pi$, satisfied by $\mathcal{A}$. Then, we transform $\Pi$ into an epistemic reduct $\Pi^\Phi$ w.r.t. $\Phi$ by replacing every $\mathtt{not}\,\mathsf{K}\,L \in \Phi$ with $\top$ and every $\mathtt{not}\,\mathsf{K}\,L \in Ep(\Pi) \setminus \Phi$ with $\mathtt{not}\,L$. Finally, $\mathcal{A}$ is a world view of $\Pi$ if

1. $\mathcal{A} = \mathtt{AS}(\Pi^\Phi) = \{A \ : \ A \text{ is an answer set of } \Pi^\Phi\}$;
2. $\Phi_{\mathcal{A}}$ agrees with $\Phi$, i.e., $\Phi_{\mathcal{A}} = \Phi$;
3. $\Phi$ is maximal, i.e., there is no bigger guess $\Phi' \supset \Phi$ such that $\mathcal{A}' = \mathtt{AS}(\Pi^{\Phi'})$ and $\Phi_{\mathcal{A}'} = \Phi'$ for some nonempty collection $\mathcal{A}'$ of consistent sets of objective literals.

Let us illustrate their approach by an important application and motivation of ES: *Closed Wold Assumption* (CWA), which says that "*$p$ is assumed to be false if there is no evidence to the contrary*" and is expressed in ASP by $\sim p \leftarrow \mathtt{not} p$[7]. However, it is formalised more adequately in ES as $\sim p \leftarrow \mathtt{not}\mathsf{M}\,p$ by [12] or $\sim p \leftarrow \mathtt{not}\mathsf{K}\,p$ by [29]. For instance, let $\Pi = \big\{\widetilde{p} \leftarrow \mathtt{not}\mathsf{K}\,p \mid \bot \leftarrow p, \widetilde{p}$. Then, take the guess $\Phi = \{\mathtt{not}\mathsf{K}\,p\}$. Thus, $\Pi^\Phi = \big\{\widetilde{p} \leftarrow \top \mid \bot \leftarrow p, \widetilde{p}$. Clearly, $\mathtt{AS}(\Pi^\Phi) = \{\{\widetilde{p}\}\}$ and $\{\{\widetilde{p}\}\} \models_{\mathsf{ES}} \Phi$. Since $\Phi$ is the maximal guess possible (see item 3 above), $\{\{\widetilde{p}\}\}$ is the unique world view of $\Pi$.

[7] However, this formalisation was then discovered to cause problems [12]. Consider $\Pi = \big\{p \,\mathtt{or}\, q \mid \sim p \leftarrow \mathtt{not} p$. Then, $\mathtt{AS}(\Pi) = \big\{\{p\}, \{q, \sim p\}\big\}$, and it answers the query $\sim p$? unknown (as it does not appear in both answer sets) while $p$ is undetermined. This result is unintended.

# 5 Our Approach: Epistemic ASP (E-ASP) and Its Epistemic Views

This section introduces an epistemic extension of ASP called *epistemic ASP*. We begin with a discussion on our motivation and the main differences with other approaches.

## 5.1 Motivation and Novelty

The problem of incomplete information in ASP still matters after more than two decades of research on the subject. Despite their successes, the approaches [4,20,29] are not fully satisfactory, and some of their seemingly intuitive results are still under discussion.

Compared to Kahl's language [18], we introduce a simpler language. We propose a modest syntax character, allowing only one epistemic operator K. However, different from ES, K may also appear in the head of a rule. We find our language strong enough to solve the problem of incomplete information in ASP because most of the critical examples in the literature, including Gelfond's $\Pi_G$ (1) and the new formalisation of CWA, use notK only to solve the quantification problem. Note that Kahl and others use also M as dual of K. Besides, we more naturally extend the syntax of ASP through the same structure of program rules, allowing not to appear only in front of literals. Note that Kahl and others use it in a literal formation as notK, notKnot, Knot etc.

The semantics of the new language is via an *epistemic view*, which is a straightforward generalisation of the answer set notion in ASP. Different from world view semantics, our semantics approach exploits a two-fold computation procedure, by splitting the program into two levels: we first look for if the candidate model, which is involved in the reduction process, is a maximal minimal model of the first level. Our reduct definition is oriented to eliminate NAF in a similar way with that of ASP. Existing reduct definitions simplify the program by removing subjective literals in the form of K$l$, M$l$, notK$l$ and notM$l$ (but, not 'not$l$') for an objective literal $l$. So, our reduct is always a positive program containing no NAF in it. The minimality condition is understood in the sense of set inclusion. It is given by checking the minimality of each set making up the (biggest possible) collection. This is similar to the method, searching for answer sets. Second, we check if such minimal models of the first level are compatible with the second level, composed of only the constraints of the main program. So, we aim to solve the recent constraint problem discussed in Sect. 2.2. We find the semantics approaches of [18,29] a bit nonstandard: first, is it a right attitude to eliminate the "positive" constructs in the form of K $l$ and M $l$ especially while world views of ES are given as kind of S5 models? Second, why do we force each valuation in such S5 models to be an answer set of the resulting program? In ES, we ask the query to the collection as a whole rather than its elements separately. At least, there is something going wrong in these approaches as always a new unintended model is being discovered, and then the reduction definitions have to be changed. To end with, our semantics approach can also be more smoothly

adapted to E-ASP programs with NAF in the head [16] and to E-ASP programs with nested expressions [23], also including subjective literal $\mathsf{K}\, l$.

### 5.2 The Language of Epistemic ASP ($\mathcal{L}_{\text{E-ASP}}$)

Literals ($\lambda$) of $\mathcal{L}_{\text{E-ASP}}$ are of two types: *objective literals ($l$)* and *subjective literals ($g$)*.

| $l$ | $g$ |
|---|---|
| $p \quad \mid \quad \sim p$ | $\mathsf{K}\, p \quad \mid \quad \mathsf{K} \sim p$ |

in which $p \in \mathbb{P}$, and $\sim$ denotes strong negation. $\mathsf{K}\, l$ is read "$l$ is known". Different from ES, we do not allow NAF to appear in a literal formation. However, NAF can precede any literal $\lambda$ in the body of a rule, and $\mathtt{not}\lambda$ means that: *there is no evidence for $\lambda$, and so, the query $\lambda$? is undetermined*. Again, different from ES, we allow $\mathsf{K}\, l$ to appear in the head of a rule. Thus, an E-ASP program is defined as a finite collection of rules

$$\lambda_1 \,\mathtt{or}\, \ldots \,\mathtt{or}\, \lambda_k \leftarrow \lambda_{k+1}, \ldots, \lambda_m, \mathtt{not}\, \lambda_{m+1}, \ldots, \mathtt{not}\, \lambda_n$$

in which $\lambda_i$'s are arbitrary (objective or subjective) literals. When we restrict $\lambda_i$'s to objective literals, the resulting program is a disjunctive logic program [14]. Hence, E-ASP rules are conservative extensions of ASP's disjunctive rules. As we follow the same structure, extensions to richer languages are straightforward via the main ASP track.

### 5.3 The Semantics of Epistemic ASP

The semantics of E-ASP is given by an *epistemic view*. Similar to a world view, it is a nonempty collection of consistent sets of objective literals. What we substantially differ is how we pick such intuitive models from among all models of an epistemic program. Let $\Pi$ be an E-ASP program. We first split $\Pi$ into two disjoint parts. The set of all constraints $r_c \in \Pi$ constitutes the upper layer ('*top*'), symbolised by $\overline{\Pi}$. This is the part of the program where we decide the ultimate epistemic views of $\Pi$ through the process: *refute*, *accept* or *reorganise*. The rest, i.e., the set $\Pi \setminus \overline{\Pi}$ forms the lower layer ('*bottom*'), where we determine the collections of possible belief sets. We denote it by $\underline{\Pi}$.

*Example 1.* Given a program

$$\Sigma = \Big\{ p \leftarrow \mathtt{not} \sim q \mid \sim q \leftarrow \mathtt{not}\, p \mid r \leftarrow \mathtt{not}\, \mathsf{K}\, p \mid \leftarrow \mathtt{not}\, r \tag{2}$$

we have $\underline{\Sigma} = \big\{ p \leftarrow \mathtt{not} \sim q \mid \sim q \leftarrow \mathtt{not}\, p \mid r \leftarrow \mathtt{not}\, \mathsf{K}\, p$ and $\overline{\Sigma} = \big\{ \leftarrow \mathtt{not}\, r$.

We start by computing the epistemic views of $\underline{\Pi}$, each of which are then involved in an evaluation process carried out in $\overline{\Pi}$. However, if $\underline{\Pi} = \emptyset$, then $\mathtt{EV}(\Pi) = \mathtt{EV}(\overline{\Pi})$, where $\mathtt{EV}(\Pi)$ denotes the set of all epistemic views of $\Pi$. In this case, the

epistemic view of the program is either $\{\emptyset\}$ or none. For instance, $\mathtt{EV}(\{\leftarrow p\}) = \big\{\{\emptyset\}\big\}$. Recall that $\leftarrow p$ has a unique answer set, namely $\emptyset$. If $\mathtt{EV}(\underline{\Pi}) = \emptyset$, then $\mathtt{EV}(\Pi) = \emptyset$. When $\overline{\Pi} = \emptyset$, $\mathtt{EV}(\Pi) = \mathtt{EV}(\underline{\Pi})$.

Our reduct based semantics is oriented to eliminate only NAF as in ASP. Remember that NAF appears as part of a construct $\mathtt{not}\,\lambda$ in an E-ASP program in which $\lambda$ is an arbitrary literal. We here follow a "guess-and-check" method: let $\mathcal{A}$ be a nonempty collection of consistent sets of objective literals, and let $A \in \mathcal{A}$. Then, $\langle \mathcal{A}, A\rangle$ is a sort of pointed (3-valued) S5 model with $A$ being the *actual* world. In an explicit representation, we simply underline the actual world $A$ in a collection $\mathcal{A}$. The partial valuation of $A$ assigns *true* to $p$ if $p \in A$ and *false* if $\sim p \in A$ (*undefined* otherwise). The reduct $\underline{\Pi}^{\langle \mathcal{A}, A\rangle}$ of $\underline{\Pi}$ w.r.t. $\langle \mathcal{A}, A\rangle$ is given by replacing every occurrence of $\mathtt{not}\,\lambda$ with[8]

**R.1** $\bot$ if $\mathcal{A}, A \models_{\text{E-ASP}} \lambda$ (simply, for $\lambda = l$ if $A \models_{\text{E-ASP}} l$; for $\lambda = \mathsf{K}\,l$ if $\mathcal{A} \models_{\text{E-ASP}} \mathsf{K}\,l$);
**R.2** $\top$ if $\mathcal{A}, A \not\models_{\text{E-ASP}} \lambda$ (simply, for $\lambda = l$ if $A \not\models_{\text{E-ASP}} l$; for $\lambda = \mathsf{K}\,l$ if $\mathcal{A} \not\models_{\text{E-ASP}} \mathsf{K}\,l$).

*Example 2.* Given a pointed model $\big\{\underline{\{p\}}, \{\sim q\}\big\}$, consider (2) above. Then, $\underline{\Sigma}^{\{\underline{\{p\}}, \{\sim q\}\}} = \big\{p \leftarrow \top \mid \sim q \leftarrow \bot \mid r \leftarrow \top$ since $\{p\} \not\models_{\text{E-ASP}} \sim q$, $\{p\} \models_{\text{E-ASP}} p$ and $\{\{p\}, \{\sim q\}\} \not\models_{\text{E-ASP}} \mathsf{K}\,p$. Now, we replace $\mathtt{notK}\,p$ by $\mathsf{K}\,p$ and $\mathtt{not}r$ by $\mathtt{notK}\,r$ in $\Sigma$ and call the resulting program $\Gamma$:

$$\Gamma = \Big\{p \leftarrow \mathtt{not} \sim q \mid \sim q \leftarrow \mathtt{not}\,p \mid r \leftarrow \mathsf{K}\,p \mid \ \leftarrow \mathtt{not}\,\mathsf{K}\,r. \tag{3}$$

Then, $\underline{\Gamma}^{\{\{p\}, \underline{\{\sim q\}}\}} = \big\{p \leftarrow \bot \mid \sim q \leftarrow \top \mid r \leftarrow \mathsf{K}\,p$ since $\{\sim q\} \models_{\text{E-ASP}} \sim q$, but $\{\sim q\} \not\models_{\text{E-ASP}} p$.

Thus, our reduct definition simplifies a program, removing only NAF w.r.t. R.1 and R.2.

First of all, we introduce a *truth-minimality* criterion, based on set inclusion over each set $A$ making up a collection $\mathcal{A}$: let $\mathbb{O}$-*Lit* be the set of all objective literals of $\mathcal{L}_{\text{E-ASP}}$, and let $\mathtt{s} : \mathcal{A} \to 2^{\mathbb{O}\text{-}Lit}$ be a (subset) map such that $\mathtt{s}(A) \subseteq A$ for every $A \in \mathcal{A}$. (When $\mathtt{s}$ equals the identity map $id$, we obtain $\mathcal{A}$ itself.) Then, a *weakening* of $\mathcal{A}$ at a point $A \in \mathcal{A}$ is identified with $\langle \mathtt{s}[\mathcal{A}], \mathtt{s}(A)\rangle$ such that $\mathtt{s} \neq id$ and $\mathtt{s}|_{\mathcal{A}\setminus\{A\}} = id$, by which we take a strict subset of $A \in \mathcal{A}$ and do not modify the rest. We say that $\langle \mathtt{s}[\mathcal{A}], \mathtt{s}(A)\rangle$ is *weaker* than $\langle \mathcal{A}, A\rangle$ and denote it by $\langle \mathtt{s}[\mathcal{A}], \mathtt{s}(A)\rangle \lhd \langle \mathcal{A}, A\rangle$. For example, the weakenings of $\{\underline{\{p, \sim q\}}, \{r\}\}$ are $\{\underline{\{p\}}, \{r\}\}$, $\{\underline{\{\sim q\}}, \{r\}\}$ and $\{\underline{\emptyset}, \{r\}\}$. Finally, we define a nonmonotonic satisfaction relation $\models^*$ for pointed (three-valued) S5 models: $\mathcal{A}, A \models^* \Pi$ if and only if

$$\mathcal{A}, A \models_{\text{E-ASP}} \Pi \text{ and } \mathtt{s}[\mathcal{A}], \mathtt{s}(A) \not\models_{\text{E-ASP}} \Pi \text{ for every } \mathtt{s} \text{ viz. } \langle \mathtt{s}[\mathcal{A}], \mathtt{s}(A)\rangle \lhd \langle \mathcal{A}, A\rangle$$

where the latter condition says that none of the weakenings of $\langle \mathcal{A}, A\rangle$ is a model of $\Pi$.

[8] The satisfaction relation $\models_{\text{E-ASP}}$ of E-ASP is the same as the relation $\models_{\text{ES}}$ (see Sect. 2.2).

**Definition 1.** *Let $\mathcal{A} \subseteq 2^{\mathbb{O}\text{-}Lit}$ be a nonempty set of consistent sets of objective literals. Then, $\mathcal{A}$ is a* minimal model *of $\Pi$ if $\mathcal{A}, A \models^* \Pi^{\langle \mathcal{A}, A\rangle}$ for every $A \in \mathcal{A}$.*

*Example 3.* $\{\{p\}, \{\sim q\}\}$ is a minimal model of $\underline{\Gamma}$ (3): $\underline{\Gamma}^{\{\underline{\{p\}}, \{\sim q\}\}} = \{p \leftarrow \mid r \leftarrow \mathsf{K}\, p$ and $\{\underline{\{p\}}, \{\sim q\}\} \models_{\mathsf{E\text{-}ASP}} \underline{\Gamma}^{\{\underline{\{p\}}, \{\sim q\}\}}$ while its weakening $\{\underline{\emptyset}, \{\sim q\}\}$ refutes it. Likewise, $\underline{\Gamma}^{\{\{p\}, \underline{\{\sim q\}}\}} = \{\sim q \leftarrow \mid r \leftarrow \mathsf{K}\, p$ and $\{\{p\}, \underline{\{\sim q\}}\} \models_{\mathsf{E\text{-}ASP}} \underline{\Gamma}^{\{\{p\}, \underline{\{\sim q\}}\}}$ while its only weakening $\{\{p\}, \underline{\emptyset}\}$ does not satisfy it. Clearly, $\{\{p, r\}\}$ and $\{\{\sim q\}\}$ are the other minimal models of $\underline{\Gamma}$. Similarly, $\{\{p, r\}, \{\sim q, r\}\}$ is a minimal model of $\underline{\Sigma}$ (2): indeed, $\underline{\Sigma}^{\{\underline{\{p,r\}}, \{\sim q, r\}\}} = \{p \leftarrow \mid r \leftarrow$ and it is obvious that $\{\underline{\{p, r\}}, \{\sim q, r\}\}$ satisfies it while all its weakenings refute it. We also have $\underline{\Sigma}^{\{\{p,r\}, \underline{\{\sim q,r\}}\}} = \{\sim q \leftarrow \mid r \leftarrow$ and $\{\{p, r\}\underline{\{\sim q, r\}}\} \models_{\mathsf{E\text{-}ASP}} \underline{\Sigma}^{\{\{p,r\}, \underline{\{\sim q,r\}}\}}$ while any of its weakenings violates it. The other two minimal models of $\underline{\Sigma}$ are $\{\{\sim q, r\}\}$ and $\{\{p\}\}$. In each program, the last two minimal models (i.e., the singleton models[9]) are unintended.

As seen above, minimality of truth does not always guarantee intuitive results. Therefore, we will now introduce a criterion to choose intended models among all such minimal models. Given an E-ASP program $\Pi$, we first define a $\Pi$-indexed partial preorder (denoted by $\preceq_\Pi$) over three-valued S5 models by: $\mathcal{A} \preceq_\Pi \mathcal{A}'$ if and only if

$$\mathcal{A} \cup \mathcal{A}', A \models_{\mathsf{E\text{-}ASP}} \Pi \text{ for all } A \in \mathcal{A} \text{ implies } \mathcal{A} \cup \mathcal{A}', A' \models_{\mathsf{E\text{-}ASP}} \Pi \text{ for all } A' \in \mathcal{A}'. \tag{4}$$

The strict version of $\preceq_\Pi$ is given as usual: $\mathcal{A} \prec_\Pi \mathcal{A}'$ iff $\mathcal{A} \preceq_\Pi \mathcal{A}'$ and $\mathcal{A}' \not\preceq_\Pi \mathcal{A}$. If $\mathcal{A} \preceq_\Pi \mathcal{A}'$ and $\mathcal{A}' \preceq_\Pi \mathcal{A}$, then $\mathcal{A}$ is *equivalent* to $\mathcal{A}'$ w.r.t. $\preceq_\Pi$ (denoted by $\mathcal{A} \approx_\Pi \mathcal{A}'$).

*Example 4.* The program $\Upsilon = \{p \,\mathtt{or}\, q \mid p \leftarrow \mathtt{notK}\, q$ has two minimal models: $\{\{p\}\}$ and $\{\{q\}\}$, among which $\{\{q\}\} \prec_r \{\{p\}\}$ since $\{\underline{\{p\}}, \{q\}\} \models_{\mathsf{E\text{-}ASP}} \Upsilon$, but $\{\{p\}, \underline{\{q\}}\} \not\models_{\mathsf{E\text{-}ASP}} \Upsilon$.

**Definition 2.** *$\mathcal{A} \subseteq 2^{\mathbb{O}\text{-}Lit}$ is an* epistemic view *of a "constraint-free" program $\Pi$ if*

1. *$\mathcal{A}$ is a minimal model of $\Pi$;*
2. *there is no minimal model $\mathcal{A}'$ of $\Pi$ such that $\mathcal{A} \prec_\Pi \mathcal{A}'$;*

[9] Singleton minimal models of a program $\Pi$ are sometimes source of a problem in capturing intuitive results: for a singleton set, $\mathsf{K}p$ and $p$ are of no difference, as well as $\mathtt{notK}p$ and $\mathtt{not}p$. Thus, an E-ASP program performs like an ASP program, and we may obtain "unjustified" minimal models. For instance, in $\underline{\Sigma}$, if we replace notK with not, the resulting ASP program has the answer sets $\{p\}$ and $\{\sim q, r\}$. Note that $\{\{p\}\}$ and $\{\{\sim q, r\}\}$ are minimal models of $\underline{\Sigma}$. We get a similar result if we change $\mathsf{K}\, p$ with $p$ in $\underline{\Gamma}$. Thus, singleton sets do not allow us to quantify over all possible beliefs. In order to overcome this obstacle, we need to check the behaviour of singletons in an interplay with other minimal models by using an ordering.

*Example 4, cont.* Thus, $\texttt{EV}(\Upsilon) = \big\{\{\{p\}\}\big\}$. Let $\Lambda = \big\{p \leftarrow \texttt{not}\, q \mid q \leftarrow \texttt{not}\mathsf{K}\, p.$ Clearly, $\{\{p\}\}$ and $\{\{q\}\}$ are the only minimal models of $\Lambda$. Recall that for singletons, $\Lambda$ behaves as an ASP program $\Lambda' = \big\{p \leftarrow \texttt{not}q \mid q \leftarrow \texttt{not}p$ and $\Lambda'$ has 2 answer sets $\{p\}$ and $\{q\}$. So, again we cannot quantify over all beliefs. Indeed, $\{\underline{\{p\}}, \{q\}\} \not\models_{\text{E-ASP}} \Lambda$, but $\{\{p\}, \underline{\{q\}}\} \models_{\text{E-ASP}} \Lambda$. Thus, we have $\{\{p\}\} \prec_\Lambda \{\{q\}\}$. Consequently, $\texttt{EV}(\Lambda) = \big\{\{\{q\}\}\big\}$.

*Example 5.* We have seen that $\underline{\Sigma}$ (2) and $\underline{\Gamma}$ (3) have 3 minimal models. Among these, we have the order, $\{\{p\}\} \prec_{\underline{\Sigma}} \{\{\sim q, r\}\} \approx_{\underline{\Sigma}} \{\{p, r\}, \{\sim q, r\}\}$ and $\{\{\sim q\}\} \approx_{\underline{\Gamma}} \{\{p, r\}\} \approx_{\underline{\Gamma}} \{\{p\}, \{\sim q\}\}$. So, the ordering $\preceq_\Pi$ is not strong enough to rule out all unintended models. When this is the case, we need to apply a third condition to compare equivalent models w.r.t. $\preceq_\Pi$.

We now introduce a *knowledge-minimising* condition: Let $\mathbb{L}(.)$ represent the set of objective literals occurring in any syntactic construct (head, body, etc). We first consider the set $\mathcal{H}_\Pi = \bigcup_{r \in \Pi} \mathbb{L}(\mathit{head}(r))$ of all objective literals occurring in the head parts of a program $\Pi$. For example, $\mathcal{H}_{\underline{\Sigma}} = \mathcal{H}_\Sigma = \mathcal{H}_\Gamma = \mathcal{H}_{\underline{\Gamma}} = \{p, \sim q, r\}$ (see (2) and (3)). Note that belief sets $A$'s of an epistemic view $\mathcal{A}$ of $\Pi$ can only contain literals from $\mathcal{H}_{\underline{\Pi}}$[10]. Inspired by [29] (but, in a different way), we define the set of all *unknowns* among the literals in $\mathcal{H}_\Pi$ w.r.t. $\mathcal{A}$ and denote it by $\Phi^\Pi_{\mathcal{A}} = \{l \in \mathcal{H}_\Pi \ : \ \mathcal{A} \models_{\text{E-ASP}} \texttt{not}\, \mathsf{K}\, l\}$.

**Definition 2, cont.**

3. $\Phi^\Pi_{\mathcal{A}}$ is maximal, i.e., there is no minimal model $\mathcal{A}'$ of $\Pi$ such that $\Phi^\Pi_{\mathcal{A}} \subset \Phi^\Pi_{\mathcal{A}'}$.

Intuitively, item 3 means $\mathcal{A}$ to answer maximum possible head-literals *undetermined*.

*Example 5, cont.* If we reconsider the above $\preceq_{\underline{\Sigma}}$-equivalent and $\preceq_{\underline{\Gamma}}$-equivalent minimal models, then we see that $\{p, \sim q\} = \Phi^{\underline{\Sigma}}_{\{\{p,r\},\{\sim q,r\}\}} \supset \Phi^{\underline{\Sigma}}_{\{\{\sim q,r\}\}} = \{p\}$. As a result, $\texttt{EV}(\underline{\Sigma}) = \big\{\{\{p, r\}, \{\sim q, r\}\}\big\}$. Similarly, $\Phi^{\underline{\Gamma}}_{\{\{p\},\{\sim q\}\}} = \{p, \sim q, r\}$, $\Phi^{\underline{\Gamma}}_{\{\{\sim q\}\}} = \{p, r\}$ and $\Phi^{\underline{\Gamma}}_{\{\{p,r\}\}} = \{\sim q\}$. Then, we have: $\Phi^{\underline{\Gamma}}_{\{\{p\},\{\sim q\}\}} \supset \Phi^{\underline{\Gamma}}_{\{\{\sim q\}\}}$ and $\Phi^{\underline{\Gamma}}_{\{\{p\},\{\sim q\}\}} \supset \Phi^{\underline{\Gamma}}_{\{\{p,r\}\}}$. Thus, $\texttt{EV}(\underline{\Gamma}) = \big\{\{\{p\}, \{\sim q\}\}\big\}$.

*Remark 1.* Note that there is also an order between the orders of item 2 and item 3: we only use item 3 over minimal models of $\Pi$ that are maximal, but equivalent w.r.t. $\preceq_\Pi$.

*Example 6.* We now consider Gelfond's program $\Pi_G$ (1): $\Pi_G$ has 3 minimal models, namely $\mathcal{A}_1 = \{\{f, i\}\}$, $\mathcal{A}_2 = \{\{h, e\}\}$ and $\mathcal{A}_3 = \{\{f, i\}, \{h, e, i\}\}$, among which $\mathcal{A}_2 \prec_\Pi \mathcal{A}_3 \approx_\Pi \mathcal{A}_1$. Thus, we need to check the unknowns of $\mathcal{A}_3$ and $\mathcal{A}_1$. Since $\{e, \sim e, h, f\} = \Phi^{\Pi_G}_{\mathcal{A}_3} \supset \Phi^{\Pi_G}_{\mathcal{A}_1} = \{e, \sim e, h\}$, we have $\texttt{EV}(\Pi_G) = \big\{\mathcal{A}_3\big\}$. However, we may not always compare maximal $\preceq_\Pi$-equivalent minimal models: let $\Omega = \big\{p \leftarrow \texttt{not}\mathsf{K}\, q \mid q \leftarrow \texttt{not}\mathsf{K}\, p.$ $\Omega$ has two minimal models $\{\{p\}\}$ and $\{\{q\}\}$ such that

[10] Fact [in ASP]: if $A \in \texttt{AS}(\Pi)$, then every $l \in A$ belongs to the head of one of the rules in $\Pi$.

$\{\{p\}\} \approx_\Omega \{\{q\}\}$ since $\{\underline{\{p\}}, \{q\}\} \not\models_{\text{E-ASP}} \Omega$ and $\{\{p\}, \underline{\{q\}}\} \not\models_{\text{E-ASP}} \Omega$. Moreover, $\Phi^{\Omega}_{\{\{p\}\}} = \{q\}$ and $\Phi^{\Omega}_{\{\{q\}\}} = \{p\}$. As a result, $\texttt{EV}(\Omega) = \big\{\{\{p\}\}, \{\{q\}\}\big\}$.

When a program $\Pi$ contains constraints, i.e, $\overline{\Pi} \neq \emptyset$, we first compute $\texttt{EV}(\underline{\Pi})$ as explained above. Then, we evaluate each $\mathcal{A} \in \texttt{EV}(\underline{\Pi})$ w.r.t. their behaviour on $\overline{\Pi}$: take $\varphi = \bigvee_{r_c \in \overline{\Pi}} body(r_c)$. For every $A \in \mathcal{A}$, if $\mathcal{A}, A \not\models_{\text{E-ASP}} \varphi$, then we *accept* $\mathcal{A}$ and call it $\mathcal{A}_{accept}$; else if $\mathcal{A}, A \models_{\text{E-ASP}} \varphi$, then we *eliminate* $\mathcal{A}$ and call it $\mathcal{A}_{refute}$. Finally, we reorganise the rest in such a way that we take the biggest possible subset $\mathcal{A}_{new} \subseteq \mathcal{A}$ such that $\mathcal{A}_{new}$ is still a minimal model of $\underline{\Pi}$ and $\mathcal{A}_{new}, A \not\models_{\text{E-ASP}} \varphi$, for every $A \in \mathcal{A}_{new}$. As a result, $\texttt{EV}(\Pi)$ is the collection of all $\mathcal{A}_{accept}$'s and $\mathcal{A}_{new}$'s. Note that when $\overline{\Pi}$ exclusively contains the constraints composed of only (negated) subjective literals, we either refute or accept the epistemic views of $\underline{\Pi}$.

*Example 5, cont.* We have seen that $\texttt{EV}(\underline{\Sigma}) = \big\{\{\{p, r\}, \{\sim q, r\}\}\big\}$ and $\texttt{EV}(\underline{\Gamma}) = \big\{\{\{p\}, \{\sim q\}\}\big\}$. As $\{\{p\}, \{\sim q\}\}$ violates $\leftarrow \texttt{notK}r$, it fails to be the epistemic view of the program $\Gamma$ (*refute*!). Hence, $\texttt{EV}(\Gamma) = \emptyset$. However, as $\{\{p, r\}, \{\sim q, r\}\} \models_{\text{E-ASP}} r$, it satisfies $\leftarrow \texttt{not}r$, and so, it passes the test (*accept*!). Thus, $\texttt{EV}(\Sigma) = \big\{\{\{p, r\}, \{\sim q, r\}\}\big\}$.

*Example 7.* Let $\Delta = \big\{p \,\texttt{or}\, q \leftarrow \;\mid r \,\texttt{or}\, s \leftarrow \texttt{notK}p \mid \;\leftarrow r$. It is easy to see that

$$\texttt{EV}(\underline{\Delta}) = \big\{\{\{p, r\}, \{q, r\}, \{p, s\}, \{q, s\}\}\big\}.$$

Then, since $\{\underline{\{p, r\}}, \underline{\{q, r\}}, \{p, s\}, \{q, s\}\} \models_{\text{E-ASP}} r$, we have to remove the actual worlds $\{p, r\}$ and $\{q, r\}$, resulting in a new collection $\{\{p, s\}, \{q, s\}\}$ (*reorganise*!). As a result, $\texttt{EV}(\Delta) = \big\{\{\{p, s\}, \{q, s\}\}\big\}$. However, while $\texttt{EV}(\underline{\Delta} \cup \{\leftarrow \texttt{K}s\}) = \texttt{EV}(\underline{\Delta})$, $\texttt{EV}(\Delta \cup \{\leftarrow \texttt{K}s\}) = \emptyset$.

*Example 8.* Let $\Psi_1 = \{\leftarrow \texttt{K}p, \texttt{not}q\}$ and $\Psi_2 = \{\leftarrow \texttt{notK}p\}$ be the one-rule (constraint) E-ASP programs. As mentioned above, the only candidate epistemic view is $\{\emptyset\}$ for $\Psi_1$ and $\Psi_2$. Since $\{\emptyset\} \not\models_{\text{E-ASP}} q$ and $\{\emptyset\} \not\models_{\text{E-ASP}} \texttt{K}\,p$, we have $\Psi_1^{\{\emptyset\}} = \{\leftarrow \texttt{K}p, \top\}$ and $\Psi_2^{\{\emptyset\}} = \{\leftarrow \top\}$. Clearly, $\texttt{EV}(\Psi_1) = \big\{\{\emptyset\}\big\}$ and $\texttt{EV}(\Psi_2) = \emptyset$.

### 5.4 Comparison of Epistemic Views with World Views and AEEMs

We here compare epistemic views with world views and AEEMs over some examples. Table 6 illustrates all these approaches. Overall, epistemic views of an E-ASP program perform well, aligning with its world views and AEEMs. However, one striking advantage of our method over existing semantics is its reasonable behaviour with programs including constraints: we have seen that $\texttt{EV}(\underline{\Gamma}) = \big\{\{\{p\}, \{\sim q\}\}\big\}$ and $\texttt{EV}(\Gamma) = \emptyset$. However, while $\underline{\Gamma}$ has a unique world view (AEEM) $\{\{p\}, \{\sim q\}\}$, when we add a constraint $\leftarrow \texttt{notK}r$ into $\underline{\Gamma}$, the resulting program $\Gamma$ has another world view (AEEM) $\{\{p, r\}\}$, violating the above property (see the last two examples of Table 6 as well).

To end with, Shen et al. [29] discuss that Pearce's equilibrium semantics suffers from circular justifications and relatedly claim that [4] inherits the same

**Table 6.** World views by both [20] and [29], AEEMs (bold), and epistemic views (bold)

| program $\Pi$ | world views | EEMs of $\Pi^*$ | epistemic views |
|---|---|---|---|
| $p \texttt{ or } q \leftarrow$ | $\{\{p\},\{q\}\}$ | $\{\{p\}\} \subset \mathbf{\{\{p\},\{q\}\}}$<br>$\{\{q\}\} \subset \mathbf{\{\{p\},\{q\}\}}$<br>$\{\{p\}\} \approx_{\Pi^*} \{\{q\}\} \approx_{\Pi^*} \{\{p\},\{q\}\}$ | $\mathbf{\{\{p\},\{q\}\}}$<br>minimal models (mm):<br>$\{\{p\}\}$ , $\{\{q\}\}$ and $\{\{p\},\{q\}\}$ |
| $p \texttt{ or } q \leftarrow$<br>$p \leftarrow \mathsf{K}\, q$ | $\{\{p\},\{q\}\}$ | $\{\{p\}\} \subset \mathbf{\{\{p\},\{q\}\}}$<br>$\{\{p\}\} \approx_{\Pi^*} \{\{p\},\{q\}\}$ | $\mathbf{\{\{p\},\{q\}\}}$<br>mm: $\{\{p\}\}$ and $\{\{p\},\{q\}\}$ |
| $p \texttt{ or } q \leftarrow$<br>$p \leftarrow \texttt{not}\, \mathsf{K}\, q$ | $\{\{p\}\}$ | $\{\{q\}\} <_{\Pi^*} \mathbf{\{\{p\}\}}$<br>(incomparable w.r.t. $\subseteq$) | $\mathbf{\{\{p\}\}}$<br>mm: $\{\{p\}\}$ |
| **CWA:** $p \texttt{ or } q \leftarrow$<br>$\widetilde{p} \leftarrow \texttt{not}\, \mathsf{K}\, p$<br>$\bot \leftarrow p, \widetilde{p}$ | $\{\{q, \widetilde{p}\}\}$ | $\{\{p\}\} <_{\Pi^*} \mathbf{\{\{q, \widetilde{p}\}\}}$<br>(incomparable w.r.t. $\subseteq$) | $\mathbf{\{\{q, \widetilde{p}\}\}}$<br>mm: $\{\{p\}\}$ and $\{\{q, \widetilde{p}\}\}$ |
| $p \texttt{ or } q \leftarrow$<br>$r \leftarrow \mathsf{K}\, p$ | $\{\{p\},\{q\}\}$ | $\{\{p,r\}\} <_{\Pi^*} \mathbf{\{\{p\},\{q\}\}}$<br>$\{\{q\}\} \subset \mathbf{\{\{p\},\{q\}\}}$<br>$\{\{q\}\} \approx_{\Pi^*} \{\{p\},\{q\}\}$ | $\mathbf{\{\{p\},\{q\}\}}$<br>mm: $\{\{q\}\}$ , $\{\{p,r\}\}$,<br>and $\{\{p\},\{q\}\}$ |
| $p \leftarrow$<br>$q \texttt{ or } r \leftarrow \mathsf{K}\, p$ | $\{\{p,q\},\{p,r\}\}$ | $\{\{p,q\}\} \subset \mathbf{\{\{p,q\},\{p,r\}\}}$<br>$\{\{p,r\}\} \subset \mathbf{\{\{p,q\},\{p,r\}\}}$<br>$\{\{p,q\}\} \approx_{\Pi^*} \{\{p,r\}\} \approx_{\Pi^*} \{\{p,q\},\{p,r\}\}$ | $\mathbf{\{\{p,q\},\{p,r\}\}}$<br>mm: $\{\{p,r\}\}$ , $\{\{p,q\}\}$,<br>and $\{\{p,q\},\{p,r\}\}$ |
| $p \leftarrow \mathsf{K}\, p$ | $\{\emptyset\}$ | $\mathbf{\{\emptyset\}}$ | $\mathbf{\{\emptyset\}}$ |
| $p \leftarrow \mathsf{K}\, p$<br>$p \leftarrow \texttt{not}\, \mathsf{K}\, p$ | none | **none** | **none**<br>no minimal models |
| $p \leftarrow \mathsf{K}\, q$<br>$q \leftarrow \mathsf{K}\, p$ | $\{\emptyset\}$ | $\mathbf{\{\emptyset\}}$ | $\mathbf{\{\emptyset\}}$<br>mm: $\{\emptyset\}$ |
| $p \leftarrow q$<br>$q \leftarrow \mathsf{K}\, p$ | $\{\emptyset\}$ | $\mathbf{\{\emptyset\}}$ | $\mathbf{\{\emptyset\}}$<br>mm: $\{\emptyset\}$ |
| $p \leftarrow \texttt{not}\, q$<br>$q \leftarrow \texttt{not}\, \mathsf{K}\, p$ | $\{\{q\}\}$ | $\{\{p\}\} <_{\Pi^*} \mathbf{\{\{q\}\}}$<br>(incomparable w.r.t. $\subseteq$) | $\mathbf{\{\{q\}\}}$<br>mm: $\{\{p\}\}$ and $\{\{q\}\}$ |
| $p \leftarrow \texttt{not}\, \mathsf{K}\, q$<br>$q \leftarrow \texttt{not}\, \mathsf{K}\, p$ | $\{\{p\}\}$ and $\{\{q\}\}$ | $\mathbf{\{\{p\}\}}$ and $\mathbf{\{\{q\}\}}$<br>(incomparable w.r.t. $\subseteq$ and $\leq_{\Pi^*}$) | $\mathbf{\{\{p\}\}}$ and $\mathbf{\{\{q\}\}}$<br>mm: $\{\{p\}\}$ and $\{\{q\}\}$ |
| $p \texttt{ or } q \leftarrow$<br>$\leftarrow \texttt{not}\, \mathsf{K}\, p$ | $\{\{p\}\}$ | $\mathbf{\{\{p\}\}}$ | **none**<br>no minimal models |
| $p \texttt{ or } q \leftarrow$<br>$r \leftarrow \texttt{not}\, \mathsf{K}\, q$<br>$\leftarrow p$ | $\{\{q\}\}$ | $\mathbf{\{\{q\}\}}$ | **none**<br>no minimal models |

circularity, leading to some undesired results. One supporting example is $\Pi = \{p \leftarrow \texttt{not}\mathsf{K}p \mid p \leftarrow p$. [29] argues that $\Pi$ has no AEEMs, but in fact, $\{\{p\}\}$ is expected to be its unique world view since $\texttt{not}\mathsf{K}p \vee p$ constitutes a tautology. However, for $\{\{p\}\}$, this formula is of no difference than $\texttt{not}p \vee p$ and it is hard to believe that the latter is a tautology. Our approach agrees with [4]: since $\{\{p\}\} \not\models_{\text{E-ASP}} \texttt{not}\mathsf{K}\, p$, we have $\Pi^{\{\{p\}\}} = \{p \leftarrow \bot \mid p \leftarrow p$. Clearly, $\{\emptyset\}$ is the unique minimal model of $\Pi^{\{\{p\}\}}$. Thus, $\texttt{EV}(\Pi) = \emptyset$. Moreover, the first rule of $\Pi$ intuitively says that *if there is no evidence for* $\mathsf{K}p$, *then* $p$ *is always true*, so as already agreed by most of the approaches in the literature, this rule does

not have a world view in ES. The second rule is just a tautology, giving no information. Under these conditions, $\Pi$ cannot have a world view in ES.

## 6 Splitting Epistemic Logic Programs

Cabalar et al. [3] have recently established a formal property called *epistemic splitting*, with which they test if a semantics proposal of ES has a reasonable behaviour when subjective literals are stratified. The idea is to separate a program $\Pi$ into two disjoint subprograms (if possible), *top* and *bottom*, among which top questions bottom via its subjective literals, and bottom never refers to head literals of top. If splitting is the case w.r.t. a set of literals $U$, then we calculate world views of $\Pi$ in four steps: first we compute the world views $\mathcal{A}_b$ of bottom; second for each $\mathcal{A}_b$, we take kind of partial reduct $\Pi_U^{\mathcal{A}_b}$ by replacing subjective literals (whose literals are included in $U$) of top with their truth values in $\mathcal{A}_b$; third we find the world views $\mathcal{A}_t$ of $\Pi_U^{\mathcal{A}_b}$, and end with a solution $\langle \mathcal{A}_b, \mathcal{A}_t \rangle$ for $\Pi$; finally we concatenate the elements of $\mathcal{A}_b$ and $\mathcal{A}_t$, and result in new world views $\mathcal{A}_b \sqcup \mathcal{A}_t = \{A_b \cup A_t \ : \ A_b \in \mathcal{A}_b \text{ and } A_t \in \mathcal{A}_t\}$, answering the queried information.

All proposed semantics trials in the literature fail to satisfy this candidate property, but Gelfond's first version [8], which suffers most, among others, the counterintuitive behaviour of cyclic programs. (Recall that [8] computes two world views $\{\emptyset\}$ and $\{\{p\}\}$ for both $p \leftarrow \mathsf{K}p$ and $p \leftarrow \mathsf{M}p$. For the former rule, while $\{\{p\}\}$ is counterintuitive, for the latter, $\{\emptyset\}$ is counterintuitive, which has been justified by almost all semantics proposals in the literature.) The other semantics that passes epistemic splitting test is Truszczyński's approach [34]. (Remember that [34] produces a world view $\{\emptyset\}$ for the program $p \leftarrow p, \mathtt{not}p$, which departs it even from ASP.) Our approach is also compatible with epistemic splitting property because first, in a splittable program we can always put all (and only) constraints composed of just (negated) subjective literals into topmost layer since they are headless, and they do not contain objective literal conjuncts, and the rest of the constraints will appear in the below layers; second, we can compute the epistemic views of the lower layers as defined in Sect. 2.2 by dividing each layer into two parts where constraints are located at the top; finally, we evaluate the final epistemic views according to the truth values of topmost subjective literals conjuncts in the candidate world view by keeping or eliminating candidate epistemic views.

*Example 9.* We can split $\Gamma$ (see (3)) into 3 layers: $L_0 = \{p \leftarrow \mathtt{not}{\sim}q \mid {\sim}q \leftarrow \mathtt{not}p$, $L_1 = \{r \leftarrow \mathsf{K}p$ and $L_2 = \{ \leftarrow \mathtt{not}\mathsf{K}r$. Then, $\mathtt{EV}(L_0) = \{\{\{p\}, \{{\sim}q\}\}\}$ and $\mathtt{EV}(L_1^{\{\{p\},\{{\sim}q\}\}}) = \mathtt{EV}(r \leftarrow \bot) = \{\{\emptyset\}\}$. Next, $\mathtt{EV}(L_0 \cup L_1) = \{\{\{p\}, \{{\sim}q\}\}\}$. Finally, since $\mathtt{EV}(L_2^{\{\{p\},\{{\sim}q\}\}}) = \mathtt{EV}(\leftarrow \top) = \emptyset$, we have $\mathtt{EV}(\Gamma) = \emptyset$.

## 7 Conclusion

In this paper, we propose a neat and more standard epistemic extension of ASP (E-ASP). E-ASP is a strong rival to existing approaches in the sense that: we

introduce a simpler and more intuitive semantics, which will be better suited for knowledge representation, and the design of intelligent agents. The new reduct definition, which is similar to that of ASP, will hopefully lead to an efficient implementation of an E-ASP program solver, allowing the new language to be of more practical use. We will search first if ASP technology can be exploited to compute epistemic views. E-ASP provides a solid framework for further language extensions of ASP. Therefore, we also plan to adapt previous language extensions of ASP to E-ASP. Finally, we would like to propose a new epistemic extension of equilibrium logic, embedding E-ASP as well.

## References


1. Baral, C., Gelfond, M.: Logic programming and knowledge representation. J. Log. Program. **19**, 73–148 (1994)
2. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, Cambridge (2001)
3. Cabalar, P., Fandinno, J., Fariñas del Cerro, L.: Splitting epistemic logic programs. In: Proceedings of the 17th International Workshop on Nonmonotonic Reasoning, NMR 2018, Tempe, Arizona, USA, 27–29 October 2018 (2018)
4. Fariñas del Cerro, L., Herzig, A., Su, E.I.: Epistemic equilibrium logic. In: Yang, Q., Wooldridge, M. (eds.) Proceedings of the 24th International Joint Conference on Artificial Intelligence, pp. 2964–2970. AAAI Press (2015). http://ijcai.org/papers15/Abstracts/IJCAI15-419.html
5. Chen, J.: The generalized logic of only knowing (GOL) that covers the notion of epistemic specifications. J. Log. Comput. **7**(2), 159–174 (1997)
6. Clark, K.L.: Negation as failure. In: Gallaire, H., Minker, J. (eds.) Logic and Data Bases, Symposium on Logic and Data Bases, Centre d'études et de recherches de Toulouse, France. Advances in Data Base Theory, pp. 293–322. Plemum Press, New York (1977)
7. Gabbay, D.M.: What is negation as failure? In: Artikis, A., Craven, R., Kesim Çiçekli, N., Sadighi, B., Stathis, K. (eds.) Logic Programs, Norms and Action. LNCS (LNAI), vol. 7360, pp. 52–78. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29414-3_5
8. Gelfond, M.: Strong introspection. In: Dean, T.L., McKeown, K. (eds.) Proceedings of the 9th National Conference on Artificial Intelligence, Anaheim, CA, USA, 14–19 July 1991, vol, 1, pp. 386–391. AAAI Press/The MIT Press (1991)
9. Gelfond, M.: Logic programming and reasoning with incomplete information. Ann. Math. Artif. Intell. **12**(1–2), 89–116 (1994)
10. Gelfond, M.: Answer sets. In: Handbook of Knowledge Representation, vol. 1, p. 285 (2008)
11. Gelfond, M.: New semantics for epistemic specifications. In: Delgrande, J.P., Faber, W. (eds.) LPNMR 2011. LNCS (LNAI), vol. 6645, pp. 260–265. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20895-9_29
12. Gelfond, M.: New definition of epistemic specifications. In: KR Seminar. Texas Tech University, 28 April 2011. (talk)
13. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Kowalski, R.A., Bowen, K.A. (eds.) Logic Programming, Proceedings of the Fifth International Conference and Symposium, Seattle, Washington, USA, 15–19 August 1988, vol. 2, pp. 1070–1080. MIT Press (1988)

14. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. New Gener. Comput. **9**(3/4), 365–386 (1991)
15. Heyting, A.: Die formalen Regeln der intuitionistischen Logik. Sitzungsber. Preuss. Akad. Wiss. **42–71**, 158–169 (1930)
16. Inoue, K., Sakama, C.: Negation as failure in the head. J. Log. Program. **35**(1), 39–78 (1998)
17. Kahl, P., Watson, R., Balai, E., Gelfond, M., Zhang, Y.: The language of epistemic specifications (refined) including a prototype solver. J. Log. Comput. (2015)
18. Kahl, P.T.: Refining the semantics for epistemic logic programs. Ph.D. thesis, Texas Tech University, Department of Computer Science, Lubblock, TX, USA, May 2014
19. Kahl, P.T., Leclerc, A.P.: Epistemic logic programs with world view constraints. In: Palù, A.D., Tarau, P., Saeedloei, N., Fodor, P. (eds.) Technical Communications of the 34th International Conference on Logic Programming, ICLP 2018, Oxford, United Kingdom, 14–17 July 2018. OpenAccess Series in Informatics OASICS, vol. 64, pp. 1:1–1:17. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2018). https://doi.org/10.4230/OASIcs.ICLP.2018.1
20. Kahl, P.T., Leclerc, A.P., Son, T.C.: A parallel memory-efficient epistemic logic program solver: harder, better, faster. CoRR abs/1608.06910 (2016). http://arxiv.org/abs/1608.06910
21. Kowalski, R.A.: Logic programming. In: Siekmann, J.H. (ed.) Computational Logic, Handbook of the History of Logic, vol. 9, pp. 523–569. Elsevier (2014). https://doi.org/10.1016/B978-0-444-51624-4.50012-5
22. Leclerc, A.P., Kahl, P.T.: A survey of advances in epistemic logic program solvers. abs/1809.07141 (2018). http://arxiv.org/abs/1809.07141. (Also in the Proceedings of the 11th International Workshop on Answer Set Programming and other Computer Paradigms, ASPOCP 2018, Oxford, UK, 18 July 2018)
23. Lifschitz, V., Tang, L.R., Turner, H.: Nested expressions in logic programs. Ann. Math. Artif. Intell. **25**(3–4), 369–389 (1999)
24. Marek, V.W., Truszczynski, M.: Stable models and an alternative logic programming paradigm. CoRR cs.LO/9809032 (1998). http://arxiv.org/abs/cs.LO/9809032
25. Niemelä, I.: Logic programs with stable model semantics as a constraint programming paradigm. Ann. Math. Artif. Intell. **25**(3–4), 241–273 (1999). https://doi.org/10.1023/A:1018930122475
26. Pearce, D.: A new logical characterisation of stable models and answer sets. In: Dix, J., Pereira, L.M., Przymusinski, T.C. (eds.) NMELP 1996. LNCS, vol. 1216, pp. 57–70. Springer, Heidelberg (1997). https://doi.org/10.1007/BFb0023801
27. Pearce, D.: Equilibrium logic. Ann. Math. Artif. Intell. **47**(1–2), 3–41 (2006)
28. Przymusinski, T.C.: On the relationship between logic programming and nonmonotonic reasoning. In: Shrobe, H.E., Mitchell, T.M., Smith, R.G. (eds.) Proceedings of the 7th National Conference on Artificial Intelligence, St. Paul, MN, USA, 21–26 August 1988, pp. 444–448. AAAI Press/The MIT Press (1988). http://www.aaai.org/Library/AAAI/1988/aaai88-078.php
29. Shen, Y., Eiter, T.: Evaluating epistemic negation in answer set programming. Artif. Intell. **237**, 115–135 (2016). https://doi.org/10.1016/j.artint.2016.04.004
30. Shen, Y., Eiter, T.: Evaluating epistemic negation in answer set programming (extended abstract). In: Sierra, C. (ed.) Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, 19–25 August 2017, pp. 5060–5064. ijcai.org (2017). https://doi.org/10.24963/ijcai.2017/722

31. Son, T.C., Le, T., Kahl, P.T., Leclerc, A.P.: On computing world views of epistemic logic programs. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, 19–25 August 2017, pp. 1269–1275 (2017). https://doi.org/10.24963/ijcai.2017/176
32. Stalnaker, R.: What is a nonmonotonic consequence relation? Fundam. Inform. **21**(1/2), 7–21 (1994)
33. Su, E.I.: Extensions of equilibrium logic by modal concepts. (Extensions de la logique d'équilibre par des concepts modaux). Ph.D. thesis, Institut de Recherche en Informatique de Toulouse, France (2015). https://tel.archives-ouvertes.fr/tel-01636791
34. Truszczyński, M.: Revisiting epistemic specifications. In: Balduccini, M., Son, T.C. (eds.) Logic Programming, Knowledge Representation, and Nonmonotonic Reasoning. LNCS (LNAI), vol. 6565, pp. 315–333. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20832-4_20
35. Wang, K., Zhang, Y.: Nested epistemic logic programs. In: Baral, C., Greco, G., Leone, N., Terracina, G. (eds.) LPNMR 2005. LNCS (LNAI), vol. 3662, pp. 279–290. Springer, Heidelberg (2005). https://doi.org/10.1007/11546207_22
36. Watson, R.: A splitting set theorem for epistemic specifications. CoRR cs.AI/0003038 (2000). http://arxiv.org/abs/cs.AI/0003038
37. Zhang, Y.: Updating epistemic logic programs. J. Log. Comput. **19**(2), 405–423 (2009). https://doi.org/10.1093/logcom/exn100
38. Zhang, Y., Zhang, Y.: Epistemic specifications and conformant planning. In: Barták, R., McCluskey, T.L., Pontelli, E. (eds.) Proceedings of the 2017 Workshop on Knowledge-Based Techniques for Problem Solving and Reasoning (KnowProS 2017) (2017)
39. Zhang, Z.: Introspecting preferences in answer set programming. In: Palù, A.D., Tarau, P., Saeedloei, N., Fodor, P. (eds.) Technical Communications of the 34th International Conference on Logic Programming, ICLP 2018, 14–17 July 2018, Oxford, United Kingdom. OASICS, vol. 64, pp. 3:1–3:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2018). https://doi.org/10.4230/OASIcs.ICLP.2018.3