Splitting Epistemic Logic Programs ∗

PEDRO CABALAR
University of Corunna, Spain
(*e-mail:* `cabalar@udc.es`)

JORGE FANDINNO
Universität Potsdam, Germany
(*e-mail:* `fandinno@uni-potsdam.de`)

LUIS FARIÑAS DEL CERRO
IRIT, University of Toulouse, CNRS, France
(*e-mail:* `farinas@irit.fr`)

submitted 1 January 2003; revised 1 January 2003; accepted 1 January 2003

Abstract

Epistemic logic programs constitute an extension of the stable model semantics to deal with new constructs called *subjective literals*. Informally speaking, a subjective literal allows checking whether some objective literal is true in all or some stable models. As it can be imagined, the associated semantics has proved to be non-trivial, since the truth of subjective literals may interfere with the set of stable models it is supposed to query. As a consequence, no clear agreement has been reached and different semantic proposals have been made in the literature. Unfortunately, comparison among these proposals has been limited to a study of their effect on individual examples, rather than identifying general properties to be checked. In this paper, we propose an extension of the well-known splitting property for logic programs to the epistemic case. We formally define when an arbitrary semantics satisfies the *epistemic splitting property* and examine some of the consequences that can be derived from that, including its relation to conformant planning and to epistemic constraints. Interestingly, we prove (through counterexamples) that most of the existing approaches fail to fulfill the epistemic splitting property, except the original semantics proposed by Gelfond in 1991 and a recent proposal by the authors, called *Founded Autoepistemic Equilibrium Logic*.

1 Introduction

The language of *epistemic specifications*, proposed by Gelfond (1991), constituted an extension of disjunctive logic programming that introduced modal operators to quantify over the set of stable models (Gelfond and Lifschitz 1988) of a program. These new constructs were later incorporated as an extension of the Answer Set Programming (ASP) paradigm in different implemented solvers (see Lecrerc and Kahl 2018b for a recent survey). The new constructs, *subjective literals*, have the form $\mathbf{K}l$ and $\mathbf{M}l$ and allow respectively checking whether an objective literal l is true in every stable model (cautious consequence) or in some stable model (brave consequence). In

∗ This is an extended version of a conference paper presented at the *Fifteenth International Conference on Logic Programming and Nonmonotonic Reasoning*, Philadelphia, PA, USA, June 3-7, 2019, where it received a best technical paper award (Cabalar et al. 2019b) An earlier version was also presented at the *Seventeenth International Workshop on Non-monotonic Reasoning* (Cabalar et al. 2018). This work was partially supported by MINECO, Spain, grant TIC2017-84453-P, Xunta de Galicia, Spain (GPC ED431B 2019/03). The second author was funded by the Centre International de Mathématiques et d'Informatique de Toulouse (CIMI) through contract ANR-11-LABEX-0040-CIMI within the program ANR-11-IDEX-0002-02 and the Alexander von Humboldt Foundation.

many cases, these subjective literals can be seen as simple queries, but what makes them really interesting is their use in rule bodies, which may obviously affect the set of stable models they are meant to quantify. This feature makes them suitable for modelling introspection (reasoning about the knowledge and lack of knowledge that the system possesses rather than reasoning exclusively about the facts themselves) but, at the same time, easily involves cyclic specifications whose intuitive behaviour is not always easy to define. For instance, the semantics of an epistemic logic program may yield alternative sets of stable models, each set being called a *world view*. Deciding the intuitive world views of a cyclic specification has motivated a wide debate in the literature. In fact, in Gelfond's original semantics (G91; Gelfond 1991) or in its extensions to arbitrary propositional formulas (Wang and Zhang 2005; Truszczyński 2011), some cyclic examples manifested self-supportedness, so Gelfond (2011) himself and, later on, other authors (Kahl et al. 2015; Fariñas del Cerro et al. 2015; Shen and Eiter 2017; Cabalar et al. 2019a) proposed different variants trying to avoid unintended results. Unfortunately, comparison among these variants was limited to studying their effect on a set of "test" examples, leading to a lack of confidence as any proposal is always subject to the appearance of new counterintuitive examples. A next methodological step would consist in defining formal properties to be established and that would cover complete families of examples and, hopefully, could help to reach an agreement on some language fragments. For instance, one would expect that, at least, the existing approaches agreed on their interpretation of acyclic specifications. Regretfully, as we will see, this is not the case.

In this paper we propose a candidate property, namely *epistemic splitting*, that not only defines an intuitive behaviour for stratified epistemic specifications but also goes further, extending the splitting theorem, well-known for standard logic programs (Lifschitz and Turner 1994), to the case of epistemic logic programs. In fact, the idea of splitting goes back to the result in (Gelfond and Przymusinska 1992) for the case of Moore's autoepistemic logic (Moore 1985). Informally speaking, we say that an epistemic logic program can be split if a part of the program (the *top*) only refers to the atoms of the other part (the *bottom*) through subjective literals. A given semantics satisfies epistemic splitting if, given any split program, it is possible to get its world views by first obtaining the world views of the bottom and then using the subjective literals in the top as "queries" on the bottom part previously obtained. If epistemic splitting holds, the semantics immediately satisfies other properties. For instance, if the use of epistemic operators is stratified, the program has a unique world view at most. Similarly, epistemic constraints (those only consisting of subjective literals) can be guaranteed to only rule out candidate world views. However, we will see that, among the previously cited approaches, only the G91 semantics and the recently proposed *Founded Autoepistemic Equilibrium Logic* (Cabalar et al. 2019a) satisfy epistemic splitting. So, somehow, most of the recent attempts to fix the behaviour of cycles have neglected the attention on the effects produced on acyclic specifications. In fact, a different property of epistemic splitting was already proved in (Watson 2000) as a method to compute world views for the G91 semantics. However, that definition was based on a "safety" condition that needed to be checked for all possible world views and is specific for G91 semantics, so it is harder to justify as a general property required for other approaches.

The rest of the paper is organised as follows. First, we motivate the main idea through a well-known example. After that, we recall basic definitions of (non-epistemic) ASP and splitting, introduce the language of epistemic specifications and define the G91 semantics. In the next section, we proceed to define the property of epistemic splitting and study some of its consequences. Then, we formally prove that G91 satisfies this property while we provide counterexamples for the approaches that do not satisfy it. Finally, before concluding the paper, we show how epistemic splitting can be applied to simplify the formalisation of conformant planning problems when representing them as epistemic logic programs.

2 Motivation

To illustrate the intuition behind our proposal, let us consider the following well-known standard example introduced in (Gelfond 1991).

Example 1. *A given college uses the following set of rules to decide whether a student X is eligible for a scholarship:*

$$\begin{aligned} \mathit{eligible}(X) &\leftarrow \mathit{high}(X) &(1)\\ \mathit{eligible}(X) &\leftarrow \mathit{minority}(X), \mathit{fair}(X) &(2)\\ {\sim}\mathit{eligible}(X) &\leftarrow {\sim}\mathit{fair}(X), {\sim}\mathit{high}(X) &(3) \end{aligned}$$

Here, '$\sim$' stands for strong negation and $\mathit{high}(X)$ and $\mathit{fair}(X)$ refer to the grades of student X. We want to encode the additional college criterion "The students whose eligibility is not determined by the college rules should be interviewed by the scholarship committee*" as another rule in the program.* □

The problem here is that, for deciding whether $\mathit{eligible}(X)$ "*can be determined*," we need to check if it holds in all the answer sets of the program, that is, if it is one of the cautious consequences of the latter. For instance, if the only available information for some student *mike* is the disjunction

$$\mathit{fair}(\mathit{mike}) \vee \mathit{high}(\mathit{mike}) \tag{4}$$

we get that program {(1)-(4)} has the following two stable models:

$$\{\mathit{high}(\mathit{mike}), \mathit{eligible}(\mathit{mike})\} \tag{5}$$

$$\{\mathit{fair}(\mathit{mike})\} \tag{6}$$

so $\mathit{eligible}(\mathit{mike})$ cannot be determined and an interview should follow. Of course, if we just want to query cautious and brave consequences of the program, we can do it inside ASP. For instance, the addition of constraint:

$$\bot \leftarrow \mathit{eligible}(\mathit{mike})$$

allows us to decide if $\mathit{eligible}(\mathit{mike})$ is a cautious consequence by just checking that the resulting program has no answer sets. The difficulty comes from the need to *derive* new information from a cautious consequence. This is where subjective literals come into play. Rule

$$\mathit{interview}(X) \leftarrow \mathtt{not}\,\mathbf{K}\mathit{eligible}(X), \mathtt{not}\,\mathbf{K}\,{\sim}\mathit{eligible}(X) \tag{7}$$

allows us to prove that $\mathit{interview}(X)$ holds whenever neither $\mathit{eligible}(X)$ nor ${\sim}\mathit{eligible}(X)$ are cautious consequences of $\{(1)\text{-}(4)\}$. Recall that $\mathbf{K}l$ holds when the literal l is true in all stable models of the program. The novel feature here is that (7) is also part of the program, and so, it affects the answer sets queried by $\mathbf{K}$ too, which would actually be:

$$\{\mathit{fair}(\mathit{mike}), \mathit{interview}(\mathit{mike})\} \tag{8}$$

$$\{\mathit{high}(\mathit{mike}), \mathit{eligible}(\mathit{mike}), \mathit{interview}(\mathit{mike})\} \tag{9}$$

So, there is a kind of cyclic reasoning: operators $\mathbf{K}$ and $\mathbf{M}$ are used to query a set of stable models that, in their turn, may depend on the application of that query. In the general case, this kind of cyclic reasoning is solved by resorting to multiple world views, but in our particular example, however, this does not seem to be needed. One would expect that separating the queried part $\{$(1)-(4)$\}$ and the rule that makes the query (7) should be correct, since the first four rules do not depend on (7) and the latter exclusively consults them without interacting with their results. A similar line of reasoning could be applied if we added one more level such as, for instance, by including the rule:

$$appointment(X) \leftarrow \mathbf{K}\, interview(X) \tag{10}$$

The two answer sets of program $\{(1)-(7)\}$ contain $interview(mike)$ and so $appointment(mike)$ can be added to both answer sets incrementally. This method of analysing a program by division into independent parts shows a strong resemblance to the *splitting theorem*, well-known for standard ASP. Splitting is applicable when the program can be divided into two parts, the *bottom* and the *top*, in such a way that the bottom never refers to head atoms in the top. When this happens, we can first compute the stable models of the bottom and then, for each one, simplify the top accordingly, getting new stable models that complete the information. We could think about different ways of extending this method for the case of epistemic logic programs, depending on how restrictive we want to be on the programs where it will be applicable. However, we will choose a very conservative case, looking for a wider agreement on the proposed behaviour. The condition we will impose is that our top program can only refer to atoms in the bottom through epistemic operators. In this way, the top is seen as a set of rules that derive facts from epistemic queries on the bottom. Thus, each world view $\mathbb{W}$ of the bottom will be used to replace the subjective literals in the top by their truth value with respect to $\mathbb{W}$. For the sake of completeness, we recall next the basic definitions of ASP and splitting, to proceed with a formalisation of epistemic splitting afterwards.

3 Background of ASP and Epistemic Specifications

Given a set of atoms At, an *objective literal* is either an atom or a truth constant[1], that is $a \in At \cup \{\top, \bot\}$, or an atom preceded by one or two occurrences of default negation, `not` a. An *objective rule* r is an implication of the form:

$$a_1 \vee \cdots \vee a_n \leftarrow L_1, \ldots, L_m \tag{11}$$

with $n \geq 0$ and $m \geq 0$, where each $a_i \in At$ is an atom and each L_j an objective literal. The left hand disjunction of (11) is called the rule *head* and abbreviated as $Head(r)$. When $n = 0$, it corresponds to $\bot$ and r is called a *constraint*. The right hand side of (11) is called the rule *body* and abbreviated as $Body(r)$. When $m = 0$, the body corresponds to $\top$ and r is called a *fact* (in this case, the body and the arrow symbol are usually omitted). An *objective program* Π is a (possibly infinite) set of objective rules. We write $Atoms(F)$ to represent the set of atoms occurring in any syntactic construct F (a literal, head, body, rule or program). By abuse of notation, we will sometimes respectively write $Head(r)$ and $Body(r)$ instead of $Atoms(Head(r))$ and $Atoms(Body(r))$ when it is clear by the context. A propositional interpretation I is a set of atoms. We assume that strong

[1] For a simpler description of program transformations, we allow truth constants where $\top$ denotes true and $\bot$ denotes false.

negation '$\sim a$' is just another atom in At and that the constraint $\bot \leftarrow a, \sim a$ is implicitly included in the program. We allow the use of variables, but understood as abbreviations of their possible ground instances. Given any syntactic construct F, we write $I \models F$ to stand for "I satisfies F" in classical propositional logic, where the commas correspond to conjunctions, '`not`' corresponds (under this interpretation) to classical negation and '$\leftarrow$' is just a reversed material implication. An interpretation I is a *(classical) model* of an objective program Π if it satisfies all its rules. The *reduct* of a program Π with respect to some propositional interpretation I, in symbols Π^I, is obtained by replacing in Π every negative literal $\texttt{not}\ a$ by $\top$ if $I \models \texttt{not}\ a$ or by $\bot$ otherwise. A propositional interpretation I is a *stable model* of a program Π iff it is a $\subseteq$-minimal model of Π^I. By SM$[\Pi]$, we denote the set of all stable models of Π. The following is a well-known property in ASP.

Property 1 (Supraclassicality). *Any stable model of any objective program* Π *is also a classical model of* Π.

We recall next the splitting theorem for ASP, beginning with the following definition.

Definition 1 (Splitting set). *A set of atoms* $U \subseteq \mathrm{At}$ *is a* splitting set *of a program* Π *if, for each rule* $r \in \Pi$, *one of the following conditions hold*

(i) $Atoms(r) \subseteq U$, *or*
(ii) $Head(r) \cap U = \emptyset$.

When this happens, we identify two disjoint subprograms, the bottom *and the* top, *respectively defined as*

$$b_U(\Pi) \stackrel{\text{def}}{=} \{\ r \in \Pi \mid Atoms(r) \subseteq U\ \} \qquad\qquad t_U(\Pi) \stackrel{\text{def}}{=} \Pi \setminus b_U(\Pi) \qquad \square$$

As an example, consider program Π_1:

$$a \quad \leftarrow \texttt{not}\ b \qquad\qquad b \leftarrow \texttt{not}\ a \tag{12}$$

$$c \vee d \leftarrow \texttt{not}\ a \qquad\qquad d \leftarrow a, \texttt{not}\ b \tag{13}$$

It is easy to see that $U = \{a, b\}$ is a splitting set that divides the program into two parts: the bottom $b_U(\Pi_1)$ containing the rules (12) and the top $t_U(\Pi_1)$ containing the rules (13). The keypoint of splitting is computing stable models of $b_U(\Pi)$ alone and using each one, I, to simplify $t_U(\Pi)$ accordingly. Given a splitting set U for Π and an interpretation $I \subseteq U$, we define the program $e_U(\Pi, I)$ as a transformation of the top program, $t_U(\Pi)$, where we replace each atom $a \in U$ from the splitting set by: $\top$ if $a \in I$ or $\bot$ otherwise. A pair $\langle I_b, I_t\rangle$ is said to be a *solution* of Π with respect to U iff I_b is a stable model of $b_U(\Pi)$ and I_t is a stable model of $e_U(\Pi, I_b)$. For instance, for Π_1, the bottom has two answer sets $\{a\}$ and $\{b\}$, so we get the respective simplifications $e_U(\Pi_1, \{a\})$:

$$c \vee d \leftarrow \texttt{not}\ \top \qquad\qquad d \leftarrow \top, \texttt{not}\ \bot$$

and $e_U(\Pi_1, \{b\})$:

$$c \vee d \leftarrow \texttt{not}\ \bot \qquad\qquad d \leftarrow \bot, \texttt{not}\ \top$$

The former has stable model $\{d\}$ so $\langle\{a\}, \{d\}\rangle$ is one solution. The latter has stable models $\{c\}$ and $\{d\}$ that yield other two solutions $\langle\{b\}, \{c\}\rangle$ and $\langle\{b\}, \{d\}\rangle$.

Theorem 1 (From Lifschitz and Turner 1994)**.** *Let U be a splitting set of program Π. A propositional interpretation $I \subseteq At$ is a stable model of Π iff there is a solution $\langle I_b, I_t \rangle$ of Π w.r.t U such that $I = I_b \cup I_t$.* □

Given the three solutions we obtained before, the splitting theorem guarantees that $\{a\} \cup \{d\}$, $\{b\} \cup \{c\}$ and $\{b\} \cup \{d\}$ are the three stable models of our example program Π_1.

One interesting observation is that any constraint r with $Atoms(r) \subseteq U$ is now included in the bottom $b_U(\Pi)$ but also satisfies condition (ii) in Def. 1 (it has no head atoms at all) and could be moved to the top $t_U(\Pi)$ instead. Having this in mind, let us provide now a relaxed definition of bottom and top programs in the following way. We say that the pair $\langle \hat{b}_U(\Pi), \hat{t}_U(\Pi) \rangle$ is an *arbitrary splitting* of program Π with respect to splitting set U if: $\hat{b}_U(\Pi) \cap \hat{t}_U(\Pi) = \emptyset$, $\hat{b}_U(\Pi) \cup \hat{t}_U(\Pi) = \Pi$, all rules in $\hat{b}_U(\Pi)$ satisfy condition (i) in Definition 1 and all rules in $\hat{t}_U(\Pi)$ satisfiy condition (ii) in Definition 1. With this definition, constraints on atoms in U can be arbitrarily placed in $\hat{b}_U(\Pi)$ or in $\hat{t}_U(\Pi)$.

Corollary 1. *Theorem 1 still holds if we define $b_U(\Pi)$ and $t_U(\Pi)$ to be any arbitrary splitting $\langle \hat{b}_U(\Pi), \hat{t}_U(\Pi) \rangle$ of program Π w.r.t. splitting set U.* □

Note also that, given an interpretation I, the solution $\langle I_b, I_t \rangle$ is unique and satisfies $I_b = I \cap U$ and $I_t = I \setminus U$. As a result, we obtain the following rewriting of Theorem 1:

Corollary 2. *Let U be a splitting set of program Π. A propositional interpretation $I \subseteq At$ is a stable model of Π iff $\langle I \cap U, I \setminus U \rangle$ is a solution of Π w.r.t U.* □

We extend now the syntax of ASP to the language of epistemic specifications. Any expression of the form $\mathbf{K}l$, $\mathbf{M}l$, $\mathtt{not}\ \mathbf{K}l$ or $\mathtt{not}\ \mathbf{M}l$, with l an objective literal, is called a *subjective literal*. We keep the same syntax for rules as in (11) except that body literals L_j can also be subjective literals now. Given rule r, we define the sets $Body_{obj}(r)$ and $Body_{sub}(r)$ respectively containing the objective and the subjective literals in $Body(r)$. We say that a rule is a *subjective constraint* if it is a constraint (that is, $Head(r) = \bot$) and its body exclusively consists of subjective literals (that is, $Body(r) = Body_{sub}(r)$).

We can define the concept of *model* of a program, in a similar way as we did for classical models in objective ASP. A *modal interpretation* $\mathcal{M} = \langle \mathbb{W}, I \rangle$ is pair where I is a propositional interpretation and $\mathbb{W} \subseteq 2^{At}$ is a non-empty set of propositional interpretations. A modal interpretation $\mathcal{M} = \langle \mathbb{W}, I \rangle$ *satisfies* a literal if one of the following conditions hold

1. $\langle \mathbb{W}, I \rangle \models \top$,
2. $\langle \mathbb{W}, I \rangle \not\models \bot$,
3. $\langle \mathbb{W}, I \rangle \models a$ if $a \in I$, for any atom $a \in At$,
4. $\langle \mathbb{W}, I \rangle \models \mathbf{K}l$ if $\langle \mathbb{W}, I' \rangle \models l$ for all $I' \in \mathbb{W}$,
5. $\langle \mathbb{W}, I \rangle \models \mathbf{M}l$ if $\langle \mathbb{W}, I' \rangle \models l$ for some $I' \in \mathbb{W}$, and
6. $\langle \mathbb{W}, I \rangle \models \mathtt{not}\ L$ if $\langle \mathbb{W}, I \rangle \not\models L$.

For a subjective literal L, $\langle \mathbb{W}, I \rangle \models L$ does not depend on I, so we sometimes write just $\mathbb{W} \models L$ instead of $\langle \mathbb{W}, I \rangle \models L$. For a rule r of the form (11), we write $\langle \mathbb{W}, I \rangle \models r$ iff either $\langle \mathbb{W}, I \rangle \models a_i$ for some $1 \leq i \leq n$ or $\langle \mathbb{W}, I \rangle \not\models L_j$ for some $1 \leq j \leq m$. We say that $\langle \mathbb{W}, I \rangle$ is a *model* of a program Π, written $\langle \mathbb{W}, I \rangle \models \Pi$, if it satisfies all its rules. Among the possible models of an epistemic logic program, all semantic approaches agree on selecting some preferred models called *world views*, each one being characterized by the $\mathbb{W}$ component. These world views satisfy a similar property to that of supraclassicality (Property 1) in non-epistemic ASP. In this case, however, rather than talking about classical models, we resort to the semantics of modal logic S5, so all world views of a program are also S5 models of the program. Formally, we say that $\mathbb{W}$ is a S5-model of a program Π, in symbols $\mathbb{W} \models \Pi$, when $\langle W, I \rangle \models \Pi$ for every $I \in \mathbb{W}$. Then, this property can be formally stated as follows:

Property 2 (Supra-S5)**.** *A semantics $\mathcal{S}$ satisfies* supra-S5 *when every $\mathcal{S}$-world view $\mathbb{W}$ of an epistemic program Π is also a S5-model of Π.* □

Notice that this definition is semantics-dependent in the sense that each alternative semantics $\mathcal{S}$ for epistemic specifications may define different $\mathcal{S}$-world views, although, to the best of our knowledge, all existing semantics satisfy supra-S5. Another property that is shared by all semantics is that, when Π is an objective ASP program (it has no modal epistemic operators) then it has a unique world view containing all the stable models of Π. We will formalize this property in the following way.

Property 3 (Supra-ASP)**.** *A semantics $\mathcal{S}$ satisfies* supra-ASP *if for any objective program Π either Π has a unique $\mathcal{S}$-world view $\mathbb{W} = \mathrm{SM}[\Pi] \neq \emptyset$ or both $\mathrm{SM}[\Pi] = \emptyset$ and Π has no world view at all.* □

Originally, some semantics like (Gelfond 1991) or (Truszczyński 2011), allowed empty world views $W = \emptyset$ when the program has no stable models, rather than leaving the program without world views. Since this feature is not really essential, we exclusively refer to non-empty world views in this paper.

We define next a useful transformation extending the idea of reduct to epistemic specifications, but generalised for a given signature.

Definition 2 (Subjective reduct)**.** *The* subjective reduct *of a program Π with respect to a set of propositional interpretations $\mathbb{W}$ and a signature $U \subseteq \mathrm{At}$, also written $\Pi^{\mathbb{W}}_{U}$, is obtained by replacing each subjective literal L with $Atoms(L) \subseteq U$ by; $\top$ if $\mathbb{W} \models L$ or by $\bot$ otherwise. When $U = At$ we just write $\Pi^{\mathbb{W}}$.* □

We use the same notation $\Pi^{\mathbb{W}}$ as for the standard reduct, but ambiguity is removed by the type of $\mathbb{W}$ (a set of interpretations now). This subjective reduct can be used to define the G91 semantics in the following way.

Definition 3 (G91-world view)**.** *A non-empty set of interpretations $\mathbb{W}$ is a* G91-world view *of an epistemic program Π if $\mathbb{W} = \mathrm{SM}[\Pi^{\mathbb{W}}]$.* □

The definitions of the other semantics studied in the paper are delayed until Section 5 where we prove one by one whether it satisfies epistemic splitting or not.

4 Epistemic splitting

We proceed now to introduce our definition of the epistemic splitting property. To do so, we begin by extending the idea of splitting set from (Lifschitz and Turner 1994).

Definition 4 (Epistemic splitting set)**.** *A set of atoms $U \subseteq \mathrm{At}$ is said to be an* epistemic splitting set *of a program Π if for any rule r in Π one of the following conditions hold*

(i) $Atoms(r) \subseteq U$,

(ii) $(Body_{obj}(r) \cup Head(r)) \cap U = \emptyset$.

We define a splitting *of* Π *as a pair* $\langle B_U(\Pi), T_U(\Pi)\rangle$ *satisfying* $B_U(\Pi) \cap T_U(\Pi) = \emptyset$ *and* $B_U(\Pi) \cup T_U(\Pi) = \Pi$, *and also that all rules in* $B_U(\Pi)$ *satisfy (i) and all rules in* $T_U(\Pi)$ *satisfy (ii).* □

With respect to the original definition of splitting set, we have replaced the condition for the top program, $Head(r) \cap U = \emptyset$, by the new condition (ii) which essentially means that the top program may only refer to atoms U in the bottom through epistemic operators. Note that this introduces a new kind of "dependence," so that, as happens with head atoms, objective literals in the body also depend on atoms in subjective literals. For instance, if $U = \{p,q\}$, the program $\Pi_1 = \{p \vee q \ , \ s \leftarrow p, \mathbf{K}q\}$ would not be splittable due to the second rule, since $s \notin U$ and we would also need the objective literal $p \notin U$. The reason for this restriction is to avoid imposing (to a potential semantics) a fixed way of evaluating p with respect to the world view $[\{p\},\{q\}]$ for the bottom.

Another observation is that we have kept the definition of $B_U(\Pi)$ and $T_U(\Pi)$ non-deterministic in the sense of the arbitrary splitting from the previous section: some rules can be arbitrarily included in one set or the other. These rules correspond to subjective constraints on atoms in U, since these are the only cases that may satisfy conditions (i) and (ii) simultaneously.

If we retake our example program $\Pi_2 = \{(1)-(7)\}$, we can see that the set U consisting of atoms $high(mike), fair(mike)$, $eligible(mike), minority(mike)$ and their corresponding strong negations is an epistemic splitting set that divides the program into a bottom $B_U(\Pi_2) = \{(1)-(4)\}$ and top part $T_U(\Pi_2) = \{(7)\}$. As in objective splitting, the idea is computing first the world views of the bottom program $B_U(\Pi)$ and for each one simplifying the corresponding subjective literals in the top program. Given an epistemic splitting set U for a program Π and a set of interpretations $\mathbb{W}$, we define $E_U(\Pi,\mathbb{W}) \stackrel{\text{def}}{=} T_U(\Pi)^{\mathbb{W}}_U$, that is, we make the subjective reduct of the top with respect to $\mathbb{W}$ and signature U.

Definition 5. *A pair* $\langle \mathbb{W}_b, \mathbb{W}_t\rangle$ *is said to be an* $\mathcal{S}$-solution *of* Π *with respect to an epistemic splitting set U if* $\mathbb{W}_b$ *is a* $\mathcal{S}$*-world view of* $B_U(\Pi)$ *and* $\mathbb{W}_t$ *is a* $\mathcal{S}$*-world view of* $E_U(\Pi,\mathbb{W}_b)$. □

As with the above properties, this definition is semantics-dependent in the sense that each alternative semantics $\mathcal{S}$ for epistemic specifications may define its own $\mathcal{S}$-solutions for a given U and Π, since it may define the selected $\mathcal{S}$-world views for a program in a different way. Back to our example, notice that $B_U(\Pi_2)$ is an objective program without epistemic operators. Thus, any semantics satisfying supra-ASP will provide $\mathbb{W}_b = [\{fair(mike)\},\{high(mike), eligible(mike)\}]$ as the unique world view for the bottom. The corresponding simplification of the top would be $E_U(\Pi_2,\mathbb{W}_b)$ containing (after grounding) the single rule $interview(mike) \leftarrow \texttt{not}\ \bot, \texttt{not}\ \bot$. Again, this program is objective and its unique world view would be $\mathbb{W}_t = [\{interview(mike)\}]$. Now, in the general case, to reconstruct the world views for the global program we define the operation:

$$\mathbb{W}_b \sqcup \mathbb{W}_t \quad = \quad \{\ I_b \cup I_t \mid I_b \in \mathbb{W}_b \text{ and } I_t \in \mathbb{W}_t\ \}$$

(remember that both the bottom and the top may produce multiple world views, depending on the program and the semantics we choose). In our example, $\mathbb{W}_b \sqcup \mathbb{W}_t$ would exactly contain the two stable models (8), (9) we saw in the introduction.

Property 4 (Epistemic splitting). *A semantics* $\mathcal{S}$ *satisfies* epistemic splitting *if for any epistemic splitting set U of any program* Π*:* $\mathbb{W}$ *is an* $\mathcal{S}$*-world view of* Π *iff there is an* $\mathcal{S}$*-solution* $\langle \mathbb{W}_b, \mathbb{W}_t\rangle$ *of* Π *with respect to U such that* $\mathbb{W} = \mathbb{W}_b \sqcup \mathbb{W}_t$. □

Back to the example, it can be easily seen that the world view we obtained in two steps is indeed the unique world view of the whole program, under any semantics satisfying epistemic splitting. Uniqueness of world view was obtained in this case because both the bottom program $B_U(\Pi_2)$ and the top, after simplification, $E_U(\Pi_2, \mathbb{W}_b)$ were objective programs and we assumed supra-ASP. In fact, as we see next, we can still get a unique world view (at most) when there are no cyclic dependences among subjective literals. This mimics the well-known result for *stratified negation* in logic programming. Let us define a modal dependence relation among atoms in a program Π so that $dep(a,b)$ is true iff there is a rule $r \in \Pi$ such that $a \in (Head(r) \cup Body_{obj}(r))$ and $b \in Body_{sub}(r)$.

Definition 6. *We say that an epistemic program* Π *is* epistemically stratified *if we can assign an integer mapping* $\lambda : \mathrm{At} \to \mathbb{N}$ *to each atom such that*

1. $\lambda(a) = \lambda(b)$ *for any rule* $r \in \Pi$ *and atoms* $a, b \in (Atoms(r) \setminus Body_{sub}(r))$,
2. $\lambda(a) > \lambda(b)$ *for any pair of atoms* a, b *satisfying* $dep(a,b)$. □

Take, for instance, the program $\Pi_3 = \{(1)-(7),(10)\}$. We can assign atoms *high*(*mike*), *fair*(*mike*), *minority*(*mike*) and *eligible*(*mike*) layer 0. Then *interview*(*mike*) could be assigned layer 1 and, finally, *appointment*(*mike*) can be located at layer 2. So, Π_3 is epistemically stratified.

Theorem 2. *Let* $\mathcal{S}$ *be any semantics satisfying supra-ASP and epistemic splitting and let* Π *be a finite, epistemically stratified program. Then, if* Π *has some* $\mathcal{S}$*-world view, this is unique.* □

Proof. Let λ be an integer mapping. We assume without loss of generality that for every $1 \leq i \leq j$, if there is some atom $a \in \mathcal{AT}$ such that $\lambda(j) = a$, then there is also some atom $b \in \mathcal{AT}$ such that $\lambda(i) = b$. Let $U'_i \overset{\text{def}}{=} \{\, a \in \mathcal{AT} \mid \lambda(a) = i \,\}$ and let

$$\Pi'_i \overset{\text{def}}{=} \{\, r \in \Pi \mid (Atoms(r) \setminus Body_{sub}(r)) \subseteq U'_i \,\}$$

and let us also define the sets $U_i \overset{\text{def}}{=} U'_1 \cup \ldots \cup U'_i$ and $\Pi_i \overset{\text{def}}{=} \Pi'_1 \cup \ldots \cup \Pi'_i$. Then, $\Pi'_1, \ldots, \Pi'_n$ is a partition of Π and, for $1 < i \leq n$, we can see that U_i is a splitting set of the program Π_i with $B_{U_i}(\Pi_i) = \Pi_{i-1}$ and $T_{U_i}(\Pi_i) = \Pi'_i$. Furthermore, since $\mathcal{S}$ satisfies epistemic splitting, it follows that $\mathbb{W}_i$ is a $\mathcal{S}$-world view of Π_i iff there is a $\mathcal{S}$-solution $\langle \mathbb{W}_{i-1}, \mathbb{W}'_i \rangle$ of Π_i with respect to U_i such that $\mathbb{W}_i = \mathbb{W}_{i-1} \sqcup \mathbb{W}'_i$. Note that, by definition of solution, we get that $\mathbb{W}_{i-1}$ is a $\mathcal{S}$-world view of $B_{U_i}(\Pi_i) = \Pi_{i-1}$ and $\mathbb{W}'_i$ is a $\mathcal{S}$-world view of $E_{U_i}(\Pi_i, \mathbb{W}_{i-1})$. Then, by induction hypothesis, we get that $\mathbb{W}_{i-1} = \mathbb{W}'_1 \sqcup \ldots \sqcup \mathbb{W}'_{i-1}$ and, thus, we get $\mathbb{W}_i = \mathbb{W}'_1 \sqcup \ldots \sqcup \mathbb{W}'_i$. As a result, it follows $\mathbb{W} = \mathbb{W}_n = \mathbb{W}'_1 \sqcup \ldots \sqcup \mathbb{W}'_n$. Furthermore, it is easy to see that $\Pi_1 = \Pi'_1$ must be an objective program (possibly empty) and, since $\mathcal{S}$ satisfies supra-ASP, we get that it has a unique $\mathcal{S}$-world view $\mathbb{W}_1$ at most. Moreover, for $1 < i \leq n$, we get that $E_{U_i}(\Pi'_i, \mathbb{W}_{i-1})$ is also an objective program and, thus, it has a unique $\mathcal{S}$-world view $\mathbb{W}'_i$ at most. Then, it follows by induction that $\mathbb{W}$ is unique, if it exists. □

The proof of the theorem just relies on multiple applications of splitting to each layer and the fact that each simplification $E_U(\Pi, \mathbb{W}_b)$ will be an objective program. This is very easy to see in the extended example Π_3. We can split the program using as U all atoms but *appointment*(*mike*) to get a bottom Π_2 and a top $\{(10)\}$. Program Π_2 can be split in its turn as we saw before, producing the unique world view $[(8),(9)]$. Then $E_U(\Pi_3, [(8),(9)])$ contains the single rule

$$appointment(mike) \leftarrow \top$$

This is also an objective program whose unique world view is $[\{appointment(mike)\}]$ and, finally, the combination of these two world views yields again a unique world view

$$[\ (8) \cup \{appointment(mike)\}\ ,\ (9) \cup \{appointment(mike)\}\ ]$$

Supra-ASP and epistemic splitting not only guarantee the existence of (at most) one world view for epistemic stratified programs but, in fact, they also force all semantics to coincide on this class of programs (Fandinno 2019, Theorem 7) – that is, either none has a world view or all of them yield the same, unique world view. Another consequence of epistemic splitting is that subjective constraints will have a monotonic behaviour. Note first that, for a subjective constraint r, we can abbreviate $\langle \mathbb{W}, I \rangle \models r$ as $\mathbb{W} \models r$ since the I component is irrelevant. Additionally, $\mathbb{W} \models r$ means that $Body(r) = Body_{sub}(r)$ is falsified, since $Head(r) = \bot$.

Property 5 (subjective constraint monotonicity)**.** *A semantics satisfies* subjective constraint monotonicity *if, for any epistemic program* Π *and any subjective constraint r,* $\mathbb{W}$ *is a world view of* $\Pi \cup \{r\}$ *iff both* $\mathbb{W}$ *is a world view of* Π *and* $\mathbb{W} \models r$. □

Theorem 3. *Epistemic splitting implies subjective constraint monotonicity.* □

Proof. Suppose we use a semantics satisfying epistemic splitting. For any program Π and any epistemic constraint r, we can always take the whole set of atoms $U = Atoms(\Pi \cup \{r\})$ as an epistemic splitting set for $\Pi' = \Pi \cup \{r\}$ and take $B_U(\Pi') = \Pi$ and $T_U(\Pi') = \{r\}$. For any world view $\mathbb{W}$ of $B_U(\Pi')$ two things may happen. A first possibility is $\mathbb{W} \models r$, and so the body of r has some false subjective literal in $\mathbb{W}$, so $E_U(\Pi', \mathbb{W})$ would be equivalent to $\bot \leftarrow \bot$. Then, the unique world view for the top would be $\mathbb{W}_t = [\emptyset]$ and $\mathbb{W} \sqcup \mathbb{W}_t = \mathbb{W}$. A second case is $\mathbb{W} \not\models r$, so all literals in the body are satisfied and $E_U(\Pi', \mathbb{W})$ would be equivalent to $\bot \leftarrow \top$ which has no world views. To sum up, we get exactly those world views $\mathbb{W}$ of Π that satisfy r. □

5 Epistemic splitting in existing semantics

In this section, we study the property of epistemic splitting for the approaches mentioned in the introduction and start by proving that the G91 semantics does satisfy this property.

5.1 Epistemic splitting in Gelfond's original semantics

We begin introducing some notation. Let $U \subseteq At$ be any set of atoms and $\overline{U} = At \setminus U$. We define the projection of a set $\mathbb{W}$ of propositional interpretations to atoms in U as $\mathbb{W}_{|U} \stackrel{\text{def}}{=} \{I \cap U \mid I \in \mathbb{W}\}$. Using these abbreviations, we can observe that:

Observation 1. *Let* $\mathbb{W}$ *be a set of propositional interpretations and* $U \subseteq \mathrm{At}$ *be a set of atoms. Then, for any subjective literal L with* $Atoms(L) = \{a\}$*:*

i) if $a \in U$*, then* $\mathbb{W} \models L$ *iff* $\mathbb{W}_{|U} \models L$*,*
ii) if $a \notin U$*, then* $\mathbb{W} \models L$ *iff* $\mathbb{W}_{|\overline{U}} \models L$*.*

Lemma 1. *Let* Π *be a program that accepts an epistemic splitting set* $U \subseteq At$ *and let* $\mathbb{W}$ *be a set of propositional interpretations. Let* $\mathbb{W}_b = \mathbb{W}_{|U}$ *and* $\mathbb{W}_t = \mathbb{W}_{|\overline{U}}$*. Then, we get*

i) $B_U(\Pi)^{\mathbb{W}} = B_U(\Pi)^{\mathbb{W}_b}$*,*
ii) $T_U(\Pi)^{\mathbb{W}} = E_U(\Pi, \mathbb{W}_b)^{\mathbb{W}_t}$*, and*

iii) $\Pi^{\mathbb{W}} = B_U(\Pi)^{\mathbb{W}_b} \cup E_U(\Pi, \mathbb{W}_b)^{\mathbb{W}_t}$.

Proof. First, since every rule $r \in B_U(\Pi)$ satisfies $Atoms(Body_{sub}(r)) \subseteq U$, from Observation 1, it follows that $B_U(\Pi)^{\mathbb{W}} = B_U(\Pi)^{\mathbb{W}_b}$. Furthemore, for any program Γ, it is easy to check that $\Gamma^{\mathbb{W}} = (\Gamma_U^{\mathbb{W}_b})^{\mathbb{W}_t}$, that is, applying the reduct with respect to $\mathbb{W}$ is the same as applying it with respect to its projection in U and, afterwards, to the remaining part. Thus, we get $T_U(\Pi)^{\mathbb{W}} = (T_U(\Pi)_U^{\mathbb{W}_b})^{\mathbb{W}_t} = E_U(\Pi, \mathbb{W}_b)^{\mathbb{W}_t}$. Finally, we have that $\Pi^{\mathbb{W}} = (B_U(\Pi) \cup T_U(\Pi))^{\mathbb{W}} = B_U(\Pi)^{\mathbb{W}} \cup T_U(\Pi)^{\mathbb{W}}$ and, thus, the result holds. □

We are now ready to prove the main theorem.

Main Theorem. *Semantics G91 satisfies epistemic splitting.* □

Proof. Let $\mathbb{W}$ be some set of propositional interpretations and let $\mathbb{W}_b = \mathbb{W}_{|U}$ and $\mathbb{W}_t = \mathbb{W}_{|\overline{U}}$. By definition, $\mathbb{W}$ is a world view of Π if and only if $\mathbb{W} = \mathrm{SM}[\Pi^{\mathbb{W}}]$. Furthermore, since U is a modal splitting set of Π, it is easy to check that U is also a regular splitting set of the regular program $\Pi^{\mathbb{W}}$. Hence, from Corollary 1, we get that $\mathbb{W}$ is a world view of Π iff

$$\mathbb{W} \;=\; \mathrm{SM}[\Pi^{\mathbb{W}}] \;=\; \left\{ \, I_b \cup I_t \;\middle|\; I_b \in \mathrm{SM}[\hat{b}_U(\Pi^{\mathbb{W}})] \text{ and } I_t \in \mathrm{SM}[\hat{e}_U(\Pi^{\mathbb{W}}, I_b)] \, \right\}$$

for some arbitrary splitting $\langle \hat{b}_U(\Pi^{\mathbb{W}}), \hat{t}_U(\Pi^{\mathbb{W}}) \rangle$. Note that all rules belonging to $B_U(\Pi)$ have all atoms from U. Hence, we take $\hat{b}_U(\Pi^{\mathbb{W}}) \stackrel{\text{def}}{=} B_U(\Pi)^{\mathbb{W}} = B_U(\Pi)^{\mathbb{W}_b}$ (Proposition 1). Similarly, we also take $\hat{t}_U(\Pi^{\mathbb{W}}) \stackrel{\text{def}}{=} T_U(\Pi)^{\mathbb{W}} = E_U(\Pi, \mathbb{W}_b)^{\mathbb{W}_t}$. Then, we get

$$\hat{e}_U(\Pi^{\mathbb{W}}, I_b) \;=\; \hat{e}_U(\hat{t}_U(\Pi^{\mathbb{W}}), I_b) \;=\; \hat{e}_U(E_U(\Pi, \mathbb{W}_b)^{\mathbb{W}_t}, I_b)$$

Notice also that no atom occurring in $E_U(\Pi, \mathbb{W}_b)^{\mathbb{W}_t}$ belongs to U, which implies that

$$\hat{e}_U(E_U(\Pi, \mathbb{W}_b)^{\mathbb{W}_t}, I_b) = E_U(\Pi, \mathbb{W}_b)^{\mathbb{W}_t}$$

Replacing above, we have that $\mathbb{W}$ is a world view of Π iff $\mathbb{W}$ is equal to

$$\{ I_b \cup I_t \mid I_b \in \mathrm{SM}[B_U(\Pi)^{\mathbb{W}_b}], I_t \in \mathrm{SM}[E_U(\Pi, \mathbb{W}_b)^{\mathbb{W}_t}] \}$$

iff

$$\mathbb{W} = \{ I_b \cup I_t \mid I_b \in \mathbb{W}'_b \text{ and } I_t \in \mathbb{W}'_t \}$$

with $\mathbb{W}'_b = \mathrm{SM}[B_U(\Pi)^{\mathbb{W}_b}]$ and $\mathbb{W}'_t = \mathrm{SM}[E_U(\Pi, \mathbb{W}_b)^{\mathbb{W}_t}]$
iff $\mathbb{W} = \mathbb{W}'_b \sqcup \mathbb{W}'_t$. Hence, it only remains to be shown that both $\mathbb{W}_b = \mathbb{W}'_b$ and $\mathbb{W}_t = \mathbb{W}'_t$ hold. Note that $I \in \mathbb{W}_b = \mathbb{W}_{|U}$ iff $I = I' \cap U$ for some $I' \in \mathbb{W}$ iff $I = (I_b \cup I_t) \cap U$ for some $I_b \in \mathbb{W}'_b$ and $I_t \in \mathbb{W}'_t$ iff $I = (I_b \cap U) \cup (I_t \cap U)$ for some $I_b \in \mathbb{W}'_b$ and $I_t \in \mathbb{W}'_t$ iff $I = I_b$ for some $I_b \in \mathbb{W}'_b$. The fact $\mathbb{W}_t = \mathbb{W}'_t$ follows in an analogous way. □

Finally, recall that (Wang and Zhang 2005) and (Truszczyński 2011) extended the G91 semantics to cover arbitrary formulas, but on the syntax of epistemic logic programs these three semantics agree. Consequently, this proof applies directly to these semantics.

Corollary 3. *The semantics by Wang and Zhang (2005) and Truszczyński (2011) for epistemic logic programs satisfy epistemic splitting.* □

5.2 Epistemic splitting in the semantics by Gelfond (2011)

As mentioned in the introduction, Gelfond (2011) revisited the semantics of epistemic logic programs in an attempt to get rid of unintended world views. This approach, which we will denote here as G11, consisted in a modification of the reduct so that positive subjective literals were not completely removed:

Definition 7 (G11-world views)**.** *Given a logic program Π, its G11-reduct with respect to a non-empty set of interpretations $\mathbb{W}$ is a program obtained by:*

1. *replacing by $\bot$ every expression $\mathbf{K}\,l$ such that $\mathbb{W} \not\models \mathbf{K}\,l$,*
2. *removing all other occurrences of subjective literals of the form $\neg\mathbf{K}\,l$, and*
3. *replacing all other occurrences of subjective literals of the form $\mathbf{K}\,l$ by l.*

A non-empty set of interpretations $\mathbb{W}$ is a G11-world view of Π iff $\mathbb{W}$ is the set of all stable models of the G11-reduct of Π with respect to $\mathbb{W}$. □

The following counterexample shows that this semantics does not satisfy epistemic splitting:

Counterexample 2. *Let Π_4 be the program containing the following two rules:*

$$a \vee b \qquad\qquad c \leftarrow \mathbf{K}a$$

Then, all semantics mentioned in this paper agree that this program has a unique world view $\mathbb{W}_4 = [\{a\},\{b\}]$. In the particular case of the G11 semantics, we can check this fact by applying the G11-reduct. Observe that the resulting program only contains the disjunction $a \vee b$ and, thus, $\mathbb{W}_4$ is a world view of program Π_4. For semantics that satisfy epistemic splitting, like G91, we can see that $U_4 = \{a,b\}$ is a splitting set of Π_4 and, thus, we can compute the unique world view $\mathbb{W}_4$ of the bottom part, $a \vee b$, and use it to simply the top part obtaining this same world view. To show that G11 does not statify epistemic splitting, let Π_5 be the program

$$a \vee b \qquad\qquad c \leftarrow \mathbf{K}a \qquad\qquad \leftarrow \texttt{not}\ c$$

obtained by adding the last constraint to our program Π_4. Note that U_4 is also a splitting set of this program dividing the program into a bottom part containing the disjunction $a \vee b$ and a top part containing the other two rules. Then, the result of simplifying the top part with respect to the bottom world view $E_U(\Pi_5, \mathbb{W}_4)$ is

$$c \leftarrow \bot \qquad\qquad \leftarrow \texttt{not}\ c$$

which has has no stable model and, thus, no world view. As a result, we cannot generate any world view by combining with $\mathbb{W}_4$ and we obtain that Π_5 has no world views at all. Since this happens for any semantics satisfying epistemic splitting and supra-ASP, it means that there is no G91-world view either. On the other hand, we can check that this program has the G11-world view $\mathbb{W}_5 = [\{a,c\}]$. To see this fact, note that the G11 reduct of Π_5 with respect to $\mathbb{W}_5$ is

$$a \vee b \qquad\qquad c \leftarrow a \qquad\qquad \leftarrow \texttt{not}\ c$$

which has a unique stable model $\{a,c\}$. To finish our example note that $\mathbb{W}_5$ was not a world view of the first program Π_4 since its G11-reduct is the program

$$a \vee b \qquad\qquad c \leftarrow a$$

which has stable models $\{a,c\}$ and $\{b\}$. This shows that, in the G11 semantics, adding a constraint that depends on a subjective literal may lead to justify that subjective literal. □

5.3 Epistemic splitting in the semantics by Kahl et al. (2015) and Shen and Eiter (2017)

A new revision of the semantics of epistemic logic programs was introduced by Kahl et al. (2015) (denoted here as K15) who proposed a new variation of the reduct as follows[2]:

Definition 8 (K15-world views)**.** *Given a logic program Π, its K15-reduct with respect to a non-empty set of interpretations $\mathbb{W}$ is a program obtained by:*

1. *replacing by $\bot$ every expression $\mathbf{K}l$ such that $\mathbb{W} \not\models \mathbf{K}l$,*
2. *replacing all other occurrences of the expression $\mathbf{K}l$ by l.*

A non-empty set of interpretations $\mathbb{W}$ is a K15-world view of Π iff $\mathbb{W}$ is the set of all stable models of the K15-reduct of Π with respect to $\mathbb{W}$. □

To show that this semantics does not satisfy epistemic splitting we can use the same program of Counterexample 2 because the K15-reduct of Π_5 with respect to $\mathbb{W}_5 = [\{a,c\}]$ is the same as its G11-reduct and, thus, we obtain the same world view violating epistemic splitting. Note that for programs where all subjective literals are positive, the G11 and K15-reducts always coincide.

A further revision, denoted S17, was later introduced by Shen and Eiter (2017). Although the original formulation of S17 was done in terms of a so-called *epistemic negation* operator, **not** l, we use here an alternative characterisation obtained by Son et al. (2017). This characterisation amounts to translating **not** l as $\texttt{not}\ \mathbf{K}l$ and selecting a class of minimal K15-world views.

Definition 9 (S17-world views)**.** *Let Π be a logic program Π and E_Π be the set of epistemic literals that contains $\texttt{not}\ \mathbf{K}l$ for every epistemic literal of the form $\mathbf{K}l$ that occurs in Π. Let $\Phi_{\mathbb{W}} \overset{\text{def}}{=} \{\ L \in E_\Pi \mid \mathbb{W} \models L\ \}$ be the subset of E_Π satisfied by non-empty set of interpretations $\mathbb{W}$. Then, a non-empty set of interpretations $\mathbb{W}$ is a S17-world view iff is a K15-world view and there is no other K15-world view $\mathbb{W}'$ such that $\Phi_{\mathbb{W}'} \supset \Phi_{\mathbb{W}}$.* □

Notice that S17 coincides with K15 when the latter yields a unique world view, since the minimisation in that case has no effect. Therefore, Counterexample 2 also applies to S17 because it has a unique K15-world view.

It is also worth to mention that Lecrerc and Kahl (2018a) have observed that the K15 and S17 semantics do not satisfy subjective constraint monotonicity and, given Theorem 3, this can also be used as another counterexample for epistemic splitting.

Counterexample 3 (From Lecrerc and Kahl 2018a)**.** *Let Π_6 be the program containing the following two rules:*

$$a \vee b \qquad\qquad \leftarrow \texttt{not}\ \mathbf{K}a$$

which has a unique K15 and S17-world view $[\{a\}]$. Note however that the program containing just $a \vee b$ is objective and has two stable models $\{a\}$ and $\{b\}$. Thus, it has the unique world view $[\{a\},\{b\}]$, which does not satisfy the subjective constraint $\leftarrow \texttt{not}\ \mathbf{K}a$. □

In other words, in the K15 and S17 semantics adding constraints intended to remove some world views may lead to new world views. Note that there may be semantics where epistemic splitting fails but, still, subjective constraint monotonicity is satisfied. In fact, the latter was proved for G11 in (Fandinno 2019, Proposition 8).

[2] We use here the alternative modal reduct introduced in the Appendix C of (Kahl et al. 2015) because of its simplicity.

5.4 Epistemic splitting in the semantics by Fariñas del Cerro et al. (2015)

Fariñas del Cerro et al. (2015) introduced a different semantics for epistemic logic programs, denoted here as F15, which is based on a combination of Equilibrium Logic (Pearce 1997) with the modal logic S5. They semantics covers arbitrary S5-formulas, but we restrict our presentation here to the syntax of epistemic logic programs.

Definition 10. *An EHT-interpretation is a pair $\langle \mathbb{W}, h \rangle$ where $\mathbb{W}$ is a non-empty set of interpretations and $h : \mathbb{W} \longrightarrow 2^{\mathcal{AT}}$ is a function mapping each interpretation T to some subset of atoms such that $h(T) \subseteq T$.* □

Intuitively, the set $\mathbb{W}$ contains the valuation in the "there" world while the set $\{ \ h(I) \mid I \in \mathbb{W} \ \}$ corresponds to the valuation in the "here" world.

Satisfaction of formulas with respect to EHT-interpretations is defined in a similar way as with respect to modal interpretations (see Section 3). An EHT-interpretation $\langle \mathbb{W}, h \rangle$ *satisfies* a literal at point $I \in \mathbb{W}$, if one of the following conditions hold

1. $\langle \mathbb{W}, h, I \rangle \models \top$,
2. $\langle \mathbb{W}, h, I \rangle \not\models \bot$,
3. $\langle \mathbb{W}, h, I \rangle \models a$ if $a \in h(I)$, for any atom $a \in At$,
4. $\langle \mathbb{W}, h, I \rangle \models \mathbf{K}\, l$ if $\langle \mathbb{W}, h, I' \rangle \models l$ for all $I' \in \mathbb{W}$,
5. $\langle \mathbb{W}, h, I \rangle \models \mathbf{M}\, l$ if $\langle \mathbb{W}, h, I' \rangle \models l$ for some $I' \in \mathbb{W}$, and
6. $\langle \mathbb{W}, h, I \rangle \models \texttt{not}\ L$ if $\langle \mathbb{W}, id, I \rangle \not\models L$.

where $id : \mathbb{W} \longrightarrow 2^{\mathcal{AT}}$ is the identity function mapping, that is, $id(T) = T$ for every $T \in \mathbb{W}$. Then, $\langle \mathbb{W}, h, I \rangle$ *satisfies* a rule r at point $I \in \mathbb{W}$, also written $\langle \mathbb{W}, h, I \rangle \models r$, iff $\langle \mathbb{W}, h, I \rangle \models L$ for every literal $L \in Body(r)$ implies $\langle \mathbb{W}, h, I \rangle \models a$ for some atom $a \in Head(r)$. We say that $\langle \mathbb{W}, h \rangle$ is an *EHT-model* of a program Π, written $\langle \mathbb{W}, h \rangle \models \Pi$, if it satisfies all its rules at all points $I \in \mathbb{W}$.

An EHT-interpretation $\langle \mathbb{W}, h \rangle$ is said to be *total on* a set $\mathbb{X} \subseteq \mathbb{W}$ iff $h(I) = I$ for every $I \in \mathbb{X}$. It is said to be *total* iff it is total on $\mathbb{W}$. Then, equilibrium models are defined as follows:

Definition 11. *A total EHT-model $\langle \mathbb{W}, id \rangle \models \Pi$ of some program Π is an* equilibrium EHT-model *iff there is no other EHT-model $\langle \mathbb{W}, h \rangle \models \Pi$ such that $h(I) \subset I$ for some $I \in \mathbb{W}$.* □

Then, the F15-world views are defined as a selection of equilibrium EHT-models. For that, we need to introduce the following definitions:

Definition 12. *Given a logic program Π, a non-empty set of interpretations and a subset $\mathbb{X} \subseteq \mathbb{W}$, we write $\mathbb{W}, \mathbb{X} \models^* \Pi$ iff the following two conditions are satisfied:*

1. *$\langle \mathbb{W}, id, I \rangle \models \Pi$ for all $I \in \mathbb{X}$, and*
2. *if $\langle \mathbb{W}, h, I \rangle \models \Pi$ for some $I \in \mathbb{W}$ such that $\langle \mathbb{W}, h \rangle$ is total on $\mathbb{W} \setminus \mathbb{X}$, then $\langle \mathbb{W}, h \rangle$ is total.* □

Then, for any two pairs of non-empty set of interpretations $\mathbb{W}$ and $\mathbb{W}'$ we write $\mathbb{W} \leq_\Pi \mathbb{W}'$ iff

$$\mathbb{W} \cup \{I\}, \mathbb{W} \models^* \Pi \quad \textit{implies} \quad \mathbb{W}' \cup \{I\}, \mathbb{W}' \models^* \Pi$$

for every I such that I belongs to some equilibrium EHT-model of Π. □

Definition 13. *Given a logic program Π, equilibrium EHT-model $\mathbb{W}$ is called an* F15-world view *iff there is no other equilibrium EHT-model $\mathbb{W}'$ such that $\mathbb{W} \subset \mathbb{W}'$ or*[3] *$\mathbb{W} <_\Pi \mathbb{W}'$.* □

[3] As usual $\mathbb{W} <_\Pi \mathbb{W}'$ stands for $\mathbb{W} \leq_\Pi \mathbb{W}'$ and $\mathbb{W} \not\leq_\Pi \mathbb{W}'$.

The following counterexample shows that this semantics does not satisfy subjective constraint monotonicity and, thus, it does not satisfy epistemic splitting either:

Counterexample 4. *Let us recall the program* Π_4 *from Counterexample 3:*

$$a \vee b \qquad\qquad \leftarrow \texttt{not}\ \mathbf{K}a$$

which has a unique equilibrium EHT-model $[\{a\}]$*, which therefore is trivially its unique F15-world view. Then, to show that the F15 semantics does not satisfy subjective constraint monotonicity we just need to show that* $[\{a\}]$ *is not an F15-world view of the program consisting only of* $a \vee b$*. Note that* $a \vee b$ *has three equilibrium EHT-models:* $[\{a\}]$, $[\{b\}]$ *and* $[\{a\},\{b\}]$*. It is easy to see that* $[\{a\},\{b\}]$ *is the unique* $\subset$*-maximal one and, thus, the unique F15-world view.* □

In other words, as happened in the K15 and S17 semantics, adding constraints intended to remove some world views may also lead to new world views in the F15 semantics.

5.5 Epistemic splitting in the semantics by Cabalar et al. (2019a)

The semantics of *Founded Autoepistemic Equilibrium Logic* (we denote here as C19) was introduced in (Cabalar et al. 2019a) as a combination of Equilibrium Logic with Moore's Autoepistemic Logic. As happened with F15 (Fariñas del Cerro et al. 2015), C19 is also applicable to the full syntax of modal logic S5. However, when we restrict ourselves to the syntax of epistemic logic programs, C19 can be alternatively defined in terms of G91 by imposing an additional condition called *foundedness*. For simplicity, we focus here in this alternative definition.

Let us start by defining $Body^+_{ob}(r)$ and $Body^+_{sub}(r)$ as the set of all positive objective literals in the body and the set of all atoms occurring in positive subjective literals, respectively. Then, we can define an *unfounded set* as follows:

Definition 14 (Unfounded set)**.** *Let* Π *be a program and* $\mathbb{W}$ *a belief view. An* unfounded set $\mathbb{S}$ *with respect to* Π *and* $\mathbb{W}$ *is a non-empty set of pairs where, for each* $\langle X,I\rangle \in \mathbb{S}$*, we have that* X *and* I *are sets of atoms and there is no rule* $r \in \Pi$ *with* $Head(r) \cap X \neq \emptyset$ *satisfying:*

1. $(\mathbb{W},I) \models Body(r)$
2. $Body^+_{ob}(r) \cap X = \emptyset$
3. $(Head(r) \setminus X) \cap I = \emptyset$
4. $Body^+_{sub}(r) \cap Y = \emptyset$ *with* $Y = \bigcup\{\ X' \mid \langle X',I'\rangle \in \mathbb{S}\ \}$. □

The definition works in a similar way to standard unfounded sets (Leone et al. 1997, Definition 3.1). In fact, the latter corresponds to the first three conditions above, except that we use $(\mathbb{W},I)$ to check $Body(r)$, as it may contain now subjective literals. Intuitively, each I represents some potential belief set (or stable model) and X is some set of atoms without a "justifying" rule, that is, there is no $r \in \Pi$ allowing a positive derivation of atoms in X. A rule like that should have a true $Body(r)$ (condition 1) but not because of positive literals in X (condition 2) and is not used to derive other head atoms outside X (condition 3). The novelty in the definition by Cabalar et al. (2019a) is the addition of condition 4: to consider r a justifying rule, we additionally require not using any positive literal $\mathbf{K}a$ in the body such that atom a also belongs to any of the unfounded components X' in $\mathbb{S}$.

Definition 15 (Founded world view)**.** *Let* Π *be a program and* $\mathbb{W}$ *be a belief view. We say that* $\mathbb{W}$ *is* unfounded *if there is some unfounded-set* $\mathbb{S}$ *s.t., for every* $\langle X,I\rangle \in \mathbb{S}$*, we have* $I \in \mathbb{W}$ *and* $X \cap I \neq \emptyset$*.* $\mathbb{W}$ *is called* founded *otherwise.* □

Now, as said before, we can define C19 in terms of G91 using the following result from (Cabalar et al. 2019a):

Theorem 4 (Main Theorem in Cabalar et al. 2019a)**.** *Given any epistemic logic program* Π*, its C19-world views are its founded G91-world views.* □

Finally, this relation to G91 was recently used in (Fandinno 2019) to prove that:

Theorem 5 (Main Theorem in Fandinno 2019)**.** *The C19 semantics for epistemic logic programs satisfies the epistemic splitting property.* □

6 Applying Epistemic Splitting to Conformant Planning

The idea of encoding the problem of finding a conformant plan as the task of obtaining a world view was first introduced by (Kahl et al. 2015). To conclude the exploration of consequences of epistemic splitting, let us consider its possible application to simplify the representation of conformant planning problems. To this aim, consider the following simple example.

Example 5. *To turn on the light in a room, we can toggle one of two lamps* l_1 *or* l_2*. In the initial state, lamp* l_1 *is plugged but we ignore the state of* l_2*. Our goal is finding a plan that guarantees we get light in the room in one step.*

A logic program that encodes this scenario for a single transition[4] could be Π_7:

$$\begin{array}{cc} plugged(l_1) & light \leftarrow toggle(L), plugged(L) \\ plugged(l_2) \vee {\sim} plugged(l_2) & \bot \leftarrow toggle(l_1), toggle(l_2) \end{array}$$

for $L \in \{l_1, l_2\}$. As we can see, $toggle(l_1)$ would constitute a conformant plan, since we obtain $light$ regardless of the initial state, while this does not happen with plan $toggle(l_2)$. In order to check whether a given sequence of actions $A_0, \dots, A_n$ is a valid conformant plan one would expect that, if we added those facts to the program, a subjective constraint should be sufficient to check that the goal holds in all the possible outcomes. In our example, we would just use:

$$\bot \leftarrow \texttt{not}\ \mathbf{K} light \tag{14}$$

and check that the program $\Pi_7 \cup \{toggle(L)\} \cup \{(14)\}$ has some world view, varying $L \in \{l_1, l_2\}$. Subjective constraint monotonicity guarantees that the addition of this "straighforward" formalisation has the expected meaning.

This method would only allow testing if the sequence of actions constitutes a conformant plan, but does not allow generating those actions. A desirable feature would be the possibility of applying the well-known ASP methodology of separating the program into three sections: generate, define and test. In our case, the "define" and the "test" sections would respectively be Π_7 and (14), but we still miss a "generate" part, capable of considering different alternative conformant plans. The problem in this case is that we cannot use a simple choice:

$$toggle(L) \vee {\sim} toggle(L)$$

because this would allow a same action to be executed in some of the stable models and not executed in others, all inside a *same* world view. Let us assume that our epistemic semantics has

[4] For simplicity, we omit time arguments or inertia, as they are not essential for the discussion.

some way to non-deterministically generate a world view in which either $\mathbf{K}a$ or $\mathbf{K}\mathtt{not}\ a$ holds using a given set of rules $\mathit{Choice}(a)$. For instance, in the G91-semantics, we could just have the rule $\mathit{Choice}(a) = \{a \leftarrow \mathtt{not}\ \mathbf{K}\mathtt{not}\ a\}$, though other semantics may have alternative ways of expressing this intended behaviour. Then, take the program Π_8 consisting of rules

$$\mathit{Choice}(\mathit{toggle}(L)) \tag{15}$$

with $L \in \{l_1, l_2\}$ plus Π_7 and (14). If our semantics satisfies epistemic splitting, it is safe to obtain the world views in three steps: generate first the alternative world views for $\mathit{toggle}(l_1)$ and $\mathit{toggle}(l_2)$ using (15), apply Π_7 and rule out those world views not satisfying the goal light in all situations using (14). To fulfill the preconditions for applying splitting, we would actually need to replace objective literal $\mathit{toggle}(L)$ by $\mathbf{K}\mathit{toggle}(L)$ in all the bodies of Π_7, but this is safe in the current context. Now, we take the bottom program to obtain 4 possible world views $\mathbb{W}_0 = [\{\mathit{toggle}(l_1)\}]$, $\mathbb{W}_1 = [\{\mathit{toggle}(l_2)\}]$, $\mathbb{W}_2 = [\{\mathit{toggle}(l_1), \mathit{toggle}(l_2)\}]$ and $\mathbb{W}_3 = [\emptyset]$. When we combine them with the top Π_7 we obtain W'_0 consisting of two stable models:

$$\{\mathit{toggle}(l_1), \mathit{plugged}(l_2), \mathit{light}, \dots\} \qquad \{\mathit{toggle}(l_1), {\sim}\mathit{plugged}(l_2), \mathit{light}, \dots\}$$

and W'_1 consisting of other two stable models:

$$\{\mathit{toggle}(l_2), \mathit{plugged}(l_2), \mathit{light}, \dots\} \qquad \{\mathit{toggle}(l_2), {\sim}\mathit{plugged}(l_2), \dots\}$$

where the latter does not contain light. Finally, constraint (14) would rule out $\mathbb{W}'_1$.

In our opinion, this representation is more natural than the representation given in (Kahl et al. 2015), whose semantics does not satisfy neither splitting nor subjective constraint monotonicity. For the sake of comparison, in that approach, the "generate" part would consist of the following three rules:

$$\mathit{toggle}(l_1) \leftarrow \mathtt{not}\ \mathbf{K}\mathtt{not}\ \mathit{toggle}(l_1) \qquad \mathit{toggle}(l_2) \leftarrow \mathtt{not}\ \mathbf{K}\mathtt{not}\ \mathit{toggle}(l_2)$$

$$\leftarrow \mathit{toggle}(l_1), \mathit{toggle}(l_2)$$

while, for the test part, the subjective constraint (14), would need to be replaced by the following pair of rules:

$$\leftarrow \mathit{light}, \mathtt{not}\ \mathbf{K}\mathit{light} \qquad \leftarrow \mathbf{K}\mathtt{not}\ \mathit{light}$$

In the K15 semantics it is not clear how to modularly define a set of rules $\mathit{Choice}(a)$ that produce the intended effect nor how to generalise it to allow concurrent actions. Note that, if we consider a generating part without the constraint, that is,

$$\mathit{toggle}(l_1) \leftarrow \mathtt{not}\ \mathbf{K}\mathtt{not}\ \mathit{toggle}(l_1) \qquad \mathit{toggle}(l_2) \leftarrow \mathtt{not}\ \mathbf{K}\mathtt{not}\ \mathit{toggle}(l_2)$$

we would obtain a unique world view $[\{\mathit{toggle}(l_1), \mathit{toggle}(l_2)\}]$, missing thus the world view $[\{\mathit{toggle}(l_1)\}]$ which corresponds to the simplest conformant plan.

7 Conclusions

We have introduced a formal property for semantics of epistemic specifications. This property, we called *epistemic splitting*, has a strong resemblance to the splitting theorem well-known for regular ASP programs. Epistemic splitting can be applied when we can divide an epistemic logic program into a bottom part for a subset U of atoms and a top part, that only refers to atoms in U

	G91	G11	F15	K15	S17	C19
Supra-S5	✓	✓	✓	✓	✓	✓
Supra-ASP	✓	✓	✓	✓	✓	✓
Subjective constraint monotonicity	✓	✓				✓
Splitting	✓					✓
Foundness						✓

Table 1. *Summary of properties in different semantics.*

through subjective literals (those using modal epistemic operators). When this happens, epistemic splitting allows obtaining the world views of the program in two steps: first, computing the world views of the bottom and, second, using each bottom world view $\mathbb{W}$ to replace subjective literals for atoms in U in the top by their truth value with respect to $\mathbb{W}$.

We have studied several consequences of epistemic splitting. For instance, if the program is stratified with respect to subjective literals then it will have a unique world view, at most. Another consequence is that constraints only consisting of subjective literals will have a monotonic behaviour, ruling out world views that satisfy the constraint body.

Our study of the main semantics in the literature has shown that only the original semantics (G91; Gelfond 1991), its generalisations to propositional formulas (Wang and Zhang 2005; Truszczyński 2011) and Founded Autoepistemic Equilibrium Logic (C19; Cabalar et al. 2019a), satisfy epistemic splitting while the rest of approaches we considered do not, as we showed with counterexamples. Table 1 is taken from (Fandinno 2019) and summarises the known results for different semantics with respect to the properties discussed through this paper plus the *foundness* property from (Cabalar et al. 2019a). Recall that most of the semantics were born motivated by the existence of self-supported world views in the G91 semantics: for instance, the program consisting of the single rule $a \leftarrow \mathbf{K}a$ yields two world views $[\emptyset]$ and $[\{a\}]$ but the latter justifies the atom a by the mere assumption of $\mathbf{K}a$ without further evidence, something that seems counterintuitive. The foundness property precisely characterises this problem in terms of unfounded sets and, as shown in the table, is currently satisfied only by C19, which in addition keeps the good behaviour of G91 with respect to epistemic splitting.

We have also explored how epistemic splitting may facilitate the simple application of the generate-define-test methodology, well-known in ASP, to the formalisation of conformant planning. Finally, as said in the introduction, a different kind of epistemic splitting had also been proved for G91 in (Watson 2000), reinforcing the idea that this semantics can be interpreted in a modular way. Notice that the sets of programs that can be split under these two definitions are incomparable.

Recall that our main motivation when defining the splitting property was to establish a minimal requirement that seemed reasonable. Still, this property was not satisfied by most semantics. On the other hand, the semantics that do satisfy this property may satisfy even stronger notions of splitting, which will be matter of future study.

Acknowledgements. We are thankful to Michael Gelfond, Richard Watson, Patrick T. Kahl, and the anonymous reviewers of the *Seventeenth International Workshop on Non-monotonic Reasoning* (NMR'18) and the *Fifteenth International Conference on Logic Programming and Nonmonotonic Reasoning* (LPNMR'19) where preliminary versions of this work were presented, for their comments and suggestions that have helped improving the paper substantially. We are also thankful to the organisers of LPNMR'19 for their support in preparing this extended version.

References

BALDUCCINI, M., LIERLER, Y., AND WOLTRAN, S., Eds. 2019. *Proceedings of the Fifteenth International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'19).* Lecture Notes in Artificial Intelligence, vol. 11481. Springer-Verlag.

CABALAR, P., FANDINNO, J., AND FARIÑAS DEL CERRO, L. 2018. Splitting epistemic logic programs. In *Proceedings of the Seventeenth International Workshop on Non-monotonic Reasoning (NMR'18).* 81–89.

CABALAR, P., FANDINNO, J., AND FARIÑAS DEL CERRO, L. 2019a. Founded world views with autoepistemic equilibrium logic. See Balduccini et al. (2019), 134–147.

CABALAR, P., FANDINNO, J., AND FARIÑAS DEL CERRO, L. 2019b. Splitting epistemic logic programs. See Balduccini et al. (2019), 120–133.

FANDINNO, J. 2019. Founded (auto)epistemic equilibrium logic satisfies epistemic splitting. *Theory and Practice of Logic Programming 19,* 5-6, 671–687.

FARIÑAS DEL CERRO, L., HERZIG, A., AND IRAZ SU, E. 2015. Epistemic equilibrium logic. In *Proceedings of the Twenty-fourth International Joint Conference on Artificial Intelligence (IJCAI'15)*, Q. Yang and M. Wooldridge, Eds. AAAI Press, 2964–2970.

GELFOND, M. 1991. Strong introspection. In *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI'91)*, T. Dean and K. McKeown, Eds. AAAI Press / The MIT Press, 386–391.

GELFOND, M. 2011. New semantics for epistemic specifications. In *Proceedings of the Eleventh International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'11)*, J. Delgrande and W. Faber, Eds. Lecture Notes in Artificial Intelligence, vol. 6645. Springer-Verlag, 260–265.

GELFOND, M. AND LIFSCHITZ, V. 1988. The stable model semantics for logic programming. In *Proceedings of the Fifth International Conference and Symposium of Logic Programming (ICLP'88)*, R. Kowalski and K. Bowen, Eds. MIT Press, 1070–1080.

GELFOND, M. AND PRZYMUSINSKA, H. 1992. On consistency and completeness of autoepistemic theories. *Fundamenta Informaticae 16,* 1, 59–92.

KAHL, P., WATSON, R., BALAI, E., GELFOND, M., AND ZHANG, Y. 2015. The language of epistemic specifications (refined) including a prototype solver. *Journal of Logic and Computation.*

LECRERC, A. AND KAHL, P. 2018a. Epistemic logic programs with world view constraints. In *Technical communications of the Thirty-forth International Conference on Logic Programming (ICLP'18).*

LECRERC, A. AND KAHL, P. 2018b. A survey of advances in epistemic logic program solvers. In *Proceedings of the Eleventh International Workshop on Answer Set Programming and other Computer Paradigms (ASPOCP'18).*

LEONE, N., RULLO, P., AND SCARCELLO, F. 1997. Disjunctive stable models: Unfounded sets, fixpoint semantics, and computation. *Information and Computation 135,* 2, 69–112.

LIFSCHITZ, V. AND TURNER, H. 1994. Splitting a logic program. In *Proceedings of the Eleventh International Conference on Logic Programming*. MIT Press, 23–37.

MOORE, R. 1985. Semantical considerations on nonmonotonic logic. *Artificial Intelligence 25*, 75–94.

PEARCE, D. 1997. A new logical characterisation of stable models and answer sets. In *Proceedings of the Sixth International Workshop on Non-Monotonic Extensions of Logic Programming (NMELP'96)*, J. Dix, L. Pereira, and T. Przymusinski, Eds. Lecture Notes in Computer Science, vol. 1216. Springer-Verlag, 57–70.

SHEN, Y. AND EITER, T. 2017. Evaluating epistemic negation in answer set programming (extended abstract). See Sierra (2017), 5060–5064.

SIERRA, C., Ed. 2017. *Proceedings of the Twenty-sixth International Joint Conference on Artificial Intelligence (IJCAI'17)*. IJCAI/AAAI Press.

SON, T., LE, T., KAHL, P., AND LECLERC, A. 2017. On computing world views of epistemic logic programs. See Sierra (2017), 1269–1275.

TRUSZCZYŃSKI, M. 2011. Revisiting epistemic specifications. In *Logic Programming, Knowledge Representation, and Nonmonotonic Reasoning - Essays Dedicated to Michael Gelfond on the Occasion of His Sixty-fifth Birthday*, M. Balduccini and T. Son, Eds. Lecture Notes in Computer Science, vol. 6565. Springer, 315–333.

WANG, K. AND ZHANG, Y. 2005. Nested epistemic logic programs. In *Proceedings of the Eighth International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'05)*, C. Baral, G. Greco, N. Leone, and G. Terracina, Eds. Lecture Notes in Artificial Intelligence, vol. 3662. Springer-Verlag, 279–290.

WATSON, R. 2000. A splitting set theorem for epistemic specifications. *Proceedings of the Eighth International Workshop on Non-Monotonic Reasoning (NMR'00) cs.AI/0003038.*