Considering Constraint Monotonicity and Foundedness in Answer Set Programming

Yi-Dong Shen[1] and **Thomas Eiter**[2]
[1]State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
[2]Institute of Logic and Computation, Vienna University of Technology (TU Wien), Favoritenstraße 9-11, A-1040 Vienna, Austria
ydshen@ios.ac.cn, eiter@kr.tuwien.ac.at

Abstract

Should the properties of constraint monotonicity and foundedness be mandatory requirements that every answer set and world view semantics must satisfy? This question is challenging and has incurred a debate in answer set programming (ASP). In this paper, we address the question by introducing natural logic programs whose expected answer sets and world views violate these properties and thus may be viewed as counter-examples to these requirements. Specifically we use instances of the *generalized strategic companies problem* for ASP benchmark competitions as concrete examples to demonstrate that the requirements of constraint monotonicity and foundedness may exclude expected answer sets for some simple disjunctive programs and world views for some epistemic specifications. In conclusion these properties should not be mandatory conditions for an answer set and world view semantics in general.

1 Introduction

Answer set programming (ASP) is a declarative problem solving paradigm specifically oriented towards modeling combinatorial search problems arising in areas of AI such as planning, reasoning about actions, diagnosis, and beyond [Gelfond and Kah, 2014; Lifschitz, 2019]. In ASP, a problem is represented by a logic program whose semantics is given by a set of intended logic models, called *stable models* [Gelfond and Lifschitz, 1988] or *answer sets* [Gelfond and Lifschitz, 1991], which correspond to solutions to the given problem.

[Gelfond and Lifschitz, 1991] introduced a class of logic programs, called *simple disjunctive programs*, which consist of rules of the form

$$A_1 \mid \cdots \mid A_k \leftarrow B_1 \wedge \cdots \wedge B_m \wedge \neg C_1 \wedge \cdots \wedge \neg C_n \quad (1)$$

where the A_i's, B_j's and C_l's are first-order atoms,[1] "|" is a disjunctive rule head operator, and "←" is an if-then rule operator. Informally, rule (1) says that *if* the body condition $B_1 \wedge \cdots \wedge B_m \wedge \neg C_1 \wedge \cdots \wedge \neg C_n$ has been inferred to be true by any other rules, *then* some A_i in the head $A_1 \mid \cdots \mid A_k$ can be inferred. Rule (1) is a *constraint* if the head is $\bot$.

[Gelfond and Lifschitz, 1991] defined the seminal answer set semantics called *GL-semantics* for a simple disjunctive program Π by means of program transformation. Given an interpretation I, the *GL-reduct* Π^I is obtained from Π by removing first all rules of form (1) containing some $C_i \in I$ and then all $\neg C_i$ from the remaining rules. Then I is an answer set of Π if I is a minimal model of Π^I. The GL-semantics has the following two properties:

(1) **Constraint monotonicity:** For any constraint C, answer sets of $\Pi \cup \{C\}$ are answer sets of Π satisfying C [Lifschitz *et al.*, 1999].

(2) **Foundedness:** Answer sets of Π contain no unfounded set [Leone *et al.*, 1997], where a set X of atoms in an interpretation I is an *unfounded set* of I if Π has no rule of form (1) with $\{A_1, \cdots, A_k\} \cap X \neq \emptyset$ satisfying (f1)-(f3):

(f1) I satisfies the rule body condition;
(f2) $\{B_1, \cdots, B_m\} \cap X = \emptyset$; and
(f3) $(\{A_1, \cdots, A_k\} \setminus X) \cap I = \emptyset$.

In order to enable representing incomplete knowledge in a logic program that has more than one answer set, [Gelfond, 1991] further extended simple disjunctive programs to epistemic specifications with epistemic modal operators $\mathbf{K}$ and $\mathbf{M}$. For a formula F and a collection $\mathcal{A}$ of interpretations, $\mathbf{K}F$ (resp. $\mathbf{M}F$) is true in $\mathcal{A}$ if F is true in *every* (resp. *some*) $I \in \mathcal{A}$. An *epistemic specification* Π consists of rules of the form

$$L_1 \mid \cdots \mid L_m \leftarrow G_1 \wedge \cdots \wedge G_n \quad (2)$$

where each L_i is an atom, and each G_j is an atom A or its negation $\neg A$ or a modal literal $\mathbf{K}A$, $\neg\mathbf{K}A$, $\mathbf{M}A$ or $\neg\mathbf{M}A$. Rule (2) is a *subjective constraint* if its head is $\bot$ and each G_j is a modal literal.

[Gelfond, 1991] defined the first world view semantics called *G91-semantics* for an epistemic specification Π as follows. Given a collection $\mathcal{A}$ of interpretations, Π is transformed into a *modal reduct* $\Pi^{\mathcal{A}}$ w.r.t. $\mathcal{A}$ by first removing all rules of form (2) containing a modal literal G that is not true in $\mathcal{A}$, then removing the remaining modal literals. $\mathcal{A}$ is a *world view* of Π if it coincides with the collection of answer sets of $\Pi^{\mathcal{A}}$ under the GL-semantics.

[1]For simplicity, we omit *strong negation* of the form $\sim A$, which can be easily compiled away [Gelfond and Lifschitz, 1991].

It turned out that the G91-semantic for epistemic specifications has both *the problem of unintended world views with recursion through* **K** and *the problem due to recursion through* **M** [Kahl, 2014]. As an example of the first problem, the epistemic specification $\Pi = \{p \leftarrow \mathbf{K}p\}$ has two world views $\mathcal{A}_1 = \{\emptyset\}$ and $\mathcal{A}_2 = \{\{p\}\}$ under the G91-semantic; however, as commented in [Kahl, 2014], $\mathcal{A}_2$ is undesired. For the second problem, $\Pi = \{p \leftarrow \mathbf{M}p\}$ has two world views $\mathcal{A}_1 = \{\emptyset\}$ and $\mathcal{A}_2 = \{\{p\}\}$ under the G91-semantics, where as commented in [Kahl, 2014], $\mathcal{A}_1$ is undesired.

Several approaches have been proposed to address the two problems [Kahl *et al.*, 2020; del Cerro *et al.*, 2015; Shen and Eiter, 2016; Su *et al.*, 2020]. For example, [Shen and Eiter, 2016] proposed to apply epistemic negation to minimize the knowledge in world views (a novel principle they named *knowledge minimization with epistemic negation*) and presented a new definition of world views, which avoids both the problem with recursion through **K** and the problem through **M**. The proposed approach is *generic* in that it can be used to extend any answer set semantics for epistemic-free programs, such as those defined in [Gelfond and Lifschitz, 1991; Pearce, 2006; Truszczynski, 2010; Faber *et al.*, 2011; Ferraris *et al.*, 2011; Shen *et al.*, 2014; Shen and Eiter, 2019], to a world view semantics for epistemic programs. This generic semantics (called *SE16-generic semantics*) has been implemented in some solvers for epistemic specifications [Kahl *et al.*, 2019; Hecher *et al.*, 2020].

Recent works [Kahl and Leclerc, 2018; Cabalar *et al.*, 2020] further extended the notions of constraint monotonicity and foundedness from simple disjunctive programs to epistemic specifications and intended to use them as criteria/intuitions to compare different world view semantics on how they comply with these properties. [Cabalar *et al.*, 2019] also defined a stronger property than constraint monotonicity, called *epistemic splitting*, intending to establish a "minimal requirement" that a world view semantics must satisfy. Unfortunately, most of the existing world view semantics, such as those defined in [Kahl *et al.*, 2020; del Cerro *et al.*, 2015; Shen and Eiter, 2016; Su *et al.*, 2020], violate these properties; see [Fandinno *et al.*, 2021] for a survey.

This strongly motivates us to evaluate the effectiveness of using these properties as criteria/intuitions to judge answer set/world view semantics. A fundamental question is:

> Should the properties of constraint monotonicity and foundedness be regarded as mandatory requirements that *every* answer set and world view semantics must satisfy?

This question is challenging and has incurred a debate in recent ASP research [Shen and Eiter, 2020; Costantini, 2021; Su, 2021].

In this paper, we address the above question by introducing natural logic programs whose expected answer sets and world views violate constraint monotonicity and foundedness and thus may be viewed as counter-examples to these requirements. Specifically we use instances of the *generalized strategic companies problem* for ASP benchmark competitions [Cadoli *et al.*, 1997; Leone *et al.*, 2006; Shen and Eiter, 2019] as concrete examples to demonstrate that these requirements may exclude expected answer sets for some simple disjunctive programs and world views for some epistemic specifications. In conclusion, these properties should not be mandatory conditions for an answer set and world view semantics in general.

2 Preliminaries

For our concerns, it suffices to consider propositional (ground) logic programs and assume a fixed propositional language $\mathcal{L}_\Sigma$ with a countable set Σ of propositional atoms. A (propositional) formula is constructed as usual from atoms using connectives $\neg$, $\wedge$, $\vee$, $\supset$, $\top$ and $\bot$, where $\top$ and $\bot$ are two 0-place logical connectives expressing $true$ and $false$, respectively. A literal is either an atom a (positive literal) or its negation $\neg a$ (negative literal). Any subset I of Σ is called an *interpretation*; I *satisfies* an atom a (or a is true in I) if $a \in I$, and I satisfies $\neg a$ (or a is false in I) if $a \notin I$. The notions of *satisfaction* and *models* of a formula in I are defined as usual in classical logic.

We use propositional formulas as building blocks to define logic programs, and further extend the language $\mathcal{L}_\Sigma$ to include the if-then rule operator $\leftarrow$, the disjunctive rule head operator $|$, and the epistemic negation operator $\mathbf{not}$. The modal operators **K** and **M** are shorthands for $\neg\mathbf{not}$ and $\mathbf{not}\,\neg$, respectively. An *epistemic negation* is of the form $\mathbf{not}\,F$, where F is a formula from $\mathcal{L}_\Sigma$. An *epistemic formula* is constructed from atoms and epistemic negations using the above connectives. We then use epistemic formulas to construct rules and logic programs.

Definition 1. *An* epistemic program *consists of rules of the form*

$$H_1 \mid \cdots \mid H_m \leftarrow B \tag{3}$$

where B and each H_i are epistemic formulas.

An epistemic program is an *epistemic specification* if every rule (3) is of form (2); it is a *simple disjunctive program* if every rule (3) is of form (1), and is an *epistemic-free program* if it contains no epistemic negation.

For a rule r of form (3) with $m = 0$, we rewrite it as a *constraint* $\bot \leftarrow B$; when B is void, we omit $\leftarrow$. Furthermore, we use $body(r)$ and $head(r)$ to refer to the body B and head $H_1 \mid \cdots \mid H_m$ of r, respectively.

For an epistemic-free program Π consisting of rules r of form (3), an interpretation I satisfies r if I satisfies some H_i in $head(r)$ once I satisfies $body(r)$; I is a model if I satisfies all rules in Π, and I is a minimal model if it is subset-minimal among all models of Π. Π is consistent if it has a model.

The definition of satisfaction and models of epistemic programs is based on a collection of interpretations.

Definition 2. *Let Π be an epistemic program and $\mathcal{A} \neq \emptyset$ be a collection of interpretations. Let $I \in \mathcal{A}$. Then*

(1) An epistemic negation $\mathbf{not}\,F$ is true in $\mathcal{A}$ if F is false in some $J \in \mathcal{A}$, and false, otherwise. I satisfies $\mathbf{not}\,F$ w.r.t. $\mathcal{A}$ if $\mathbf{not}\,F$ is true in $\mathcal{A}$.

(2) I satisfies an epistemic formula E w.r.t. $\mathcal{A}$ if I satisfies $E^{\mathcal{A}}$ as in propositional logic, where $E^{\mathcal{A}}$ is E with every epistemic negation $\mathbf{not}\,F$ that is true in $\mathcal{A}$ replaced by $\top$, and the other epistemic negations replaced by $\bot$.

(3) I satisfies a rule r of form (3) w.r.t. $\mathcal{A}$ if I satisfies some H_i in $head(r)$ w.r.t. $\mathcal{A}$ once I satisfies $body(r)$ w.r.t. $\mathcal{A}$.

(4) $\mathcal{A}$ satisfies a rule r (or r is true in $\mathcal{A}$) if every $J \in \mathcal{A}$ satisfies r w.r.t. $\mathcal{A}$. $\mathcal{A}$ satisfies Π, or $\mathcal{A}$ is an epistemic model *of Π, if $\mathcal{A}$ satisfies all rules in Π. Π is* consistent *if it has an epistemic model.*

Thus, as $\mathbf{K}$ stands for $\neg\mathbf{not}$ and $\mathbf{M}$ for $\mathbf{not}\,\neg$, $\mathbf{K}F$ (resp. $\mathbf{M}F$) is true in $\mathcal{A}$ if F is true in every (resp. some) $J \in \mathcal{A}$.

An *answer set semantics* for an epistemic-free program Π assigns to Π a collection of interpretations called *answer sets*, which can be built on the basis of rules of Π. As described in [Gelfond, 2008], the construction of such an answer set I is guided by the following two informal principles:

(P1) I must satisfy all rules of Π; and

(P2) it should adhere to the *rationality principle*, i.e., one shall not believe anything one is not forced to believe.

Note that the principle (P1) amounts to that I must be a model of Π, while the rationality principle is used for knowledge minimization in nonmonotonic reasoning. We refer to (P1) and (P2) as the *Gelfond ASP principles*. Examples of answer set semantics following the two principles include the GL-semantics by [Gelfond and Lifschitz, 1991] for simple disjunctive programs and the DI-semantics by [Shen and Eiter, 2019] for arbitrary epistemic-free programs.

A *world view semantics* for an epistemic program Π is defined in terms of some intended epistemic models of Π, called *world views*. Examples of world view semantics include the G91-semantics by [Gelfond, 1991] for epistemic specifications and the SE16-generic semantics by [Shen and Eiter, 2016] for arbitrary epistemic programs.

3 Considering Constraint Monotonicity and Foundedness for Simple Disjunctive Programs

As mentioned in the introduction, the GL-semantics satisfies the properties of constraint monotonicity and foundedness. Before answering the question whether these properties should be mandatory conditions for answer set semantics for simple disjunctive programs, we first use an example to show limitations of the GL-semantics. [Shen and Eiter, 2019] observed that the requirement of the GL-semantics, i.e., an answer set should be a minimal model of rules of the GL-reduct, may sometimes be too strong, and as demonstrated in the following example some expected answer set may not be a minimal model of the GL-reduct.

Example 1 (generalized strategic companies (GSC) problem)**.** *Consider a generalization of the well-known strategic companies problem [Cadoli* et al.*, 1997; Leone* et al.*, 2006], which is popular for ASP benchmark competitions. Suppose a holding has companies $C = \{c_1, \ldots, c_m\}$, and it produces goods $G = \{g_1, ..., g_n\}$, where each company $c_i \in C$ produces some goods $G_i \subseteq G$. The holding wants to sell some of its companies subject to the following conditions: all products should be still in the portfolio and no company c_i for which a strategy rationale, expressed by justifications $\sigma^i_1, \ldots, \sigma^i_{k_i}$ with each $\sigma^i_j = \ell^i_1 \wedge \cdots \wedge \ell^i_{l_i}$ being a conjunction of literals on C, holds true is sold.*

The GSC problem *is to find a set $C' \subseteq C$ of companies, called a strategic set, satisfying the following four conditions:*

(1) The companies in C' produce all goods in G;

(2) C' covers all $c_i \in C$ such that some justification σ^i_j holds true relative to C';

(3) Every $c_i \in C'$ either produces some goods in G or has some justification σ^i_j that is true relative to C';

(4) C' is subset-minimal w.r.t. conditions (1)-(3).

For illustration, consider an instance of the GSC problem, where we have companies $C = \{c_1, c_2, c_3\}$, goods $G = \{g_1, g_2\}$, and the strategy conditions:

$$\sigma^1_1 = c_2 \wedge c_3,\ \sigma^2_1 = c_3 \ \text{ and } \ \sigma^3_1 = c_1 \wedge \neg c_2.$$

Suppose that g_1 can be produced either by c_1 or c_2, and g_2 produced either by c_1 or c_3.

We may naturally encode this GSC problem as the following simple disjunctive program (though other encodings are possible, this seems to be the most concise one; see [Shen and Eiter, 2019, Appendix A] for a discussion):

$$\begin{array}{lll}
\Pi: & g_1 & (1)\\
& g_2 & (2)\\
& c_1 \mid c_2 \leftarrow g_1 & \text{// } g_1 \text{ is produced either by } c_1 \text{ or } c_2 \quad (3)\\
& c_1 \mid c_3 \leftarrow g_2 & \text{// } g_2 \text{ is produced either by } c_1 \text{ or } c_3 \quad (4)\\
& c_1 \leftarrow c_2 \wedge c_3 & \text{// strategy condition } \sigma^1_1 = c_2 \wedge c_3 \quad (5)\\
& c_2 \leftarrow c_3 & \text{// strategy condition } \sigma^2_1 = c_3 \quad (6)\\
& c_3 \leftarrow c_1 \wedge \neg c_2 & \text{// strategy condition } \sigma^3_1 = c_1 \wedge \neg c_2 \quad (7)
\end{array}$$

Then, finding a strategic set $C' \subseteq C$ for this GSC problem corresponds to constructing an answer set I from Π corresponding to C', where: condition (1) for the strategic set C' corresponds to that the answer set I contains some c_i from the head of rule (3) and some c_j from the head of rule (4); condition (2) for C' corresponds to that I covers all c_i in the heads of rules (5)-(7) whose bodies are σ^i_j that is satisfied by I; condition (3) for C' corresponds to that every $c_i \in I$ is inferred from a head of some rule in Π whose body condition is satisfied by I; and condition (4) for C' corresponds to that I is subset-minimal, i.e., no proper subset J of I can be constructed in the above way.

Let us check whether $C' = \{c_1, c_2\}$ is a strategic set. First, suppose that g_1 is produced by c_2 and g_2 produced by c_1. Then C' produces all goods in G and condition (1) for C' to be a strategic set is satisfied. Corresponding to this, for answer set construction, as g_1 and g_2 are true in rules (1) and (2), the body conditions of rules (3) and (4) hold true and by choosing c_2 from the head of rule (3) and c_1 from rule (4), we infer the set $I = \{g_1, g_2, c_1, c_2\}$ from rules (1)-(4).

Second, as none of the three strategy conditions σ^1_1, σ^2_1 and σ^3_1 holds true relative to C', condition (2) for C' to be a strategic set is also satisfied. Corresponding to this, as none of the body conditions of the three rules (5)-(7) is true in I, we infer the same set $I = \{g_1, g_2, c_1, c_2\}$ from rules (1)-(7). As I satisfies all rules of Π, *it is a* candidate answer set *of* Π.

Third, as every $c_i \in C'$ produces some goods in G, condition (3) for C' to be a strategic set is satisfied. Correspondingly, every $c_i \in I$ has a rule in Π whose head contains c_i and whose body condition holds true in I.

Finally, it is easy to check that C' is subset-minimal w.r.t. conditions (1)-(3). Thus C' is a strategic set of this GSC problem. As no proper subset of I is a candidate answer set of Π as constructed in the above way, $I = \{g_1, g_2, c_1, c_2\}$ is expected to be an answer set of Π, which corresponds to the strategic set C'. One can check that $C' = \{c_1, c_2\}$ is the only strategic set for this GSC problem, and that $I = \{g_1, g_2, c_1, c_2\}$ is the only expected answer set of Π.

However, this simple disjunctive program for the GSC problem has no answer set under the GL-semantics. For $I = \{g_1, g_2, c_1, c_2\}$, the GL-reduct Π^I w.r.t. I is Π with rule (7) removed. As $\{g_1, g_2, c_1\} \subset I$ is a minimal model of Π^I, I is not a minimal model of the GL-reduct and thus is not an answer set of Π in GL-semantics.

To address issues of the GL-semantics with disjunction, [Shen and Eiter, 2019] presented an alternative answer set semantics, called *determining inference semantics* (DI-semantics for short). Briefly, for an epistemic-free program Π, the DI-semantics follows the two Gelfond ASP principles (P1) and (P2) to construct an answer set I as follows. For (P1), that I satisfies every rule of Π, the DI-semantics interprets the operator "|" in rule heads truly as a nondeterministic inference operator that returns for I from each rule $A_1 \mid \cdots \mid A_k \leftarrow Body$ of Π one A_i of the alternatives $\{A_1, \cdots, A_k\}$ once $Body$ has been inferred to be true.[2] This would produce different *candidate answer sets* for Π (corresponding to different choices of the alternatives) satisfying the principle (P1). As for (P2), that we shall not believe anything we are not forced to believe, the DI-semantics admits as answer sets only those candidate answer sets that are subset-minimal. It has been proved that for simple disjunctive programs, an answer set in GL-semantics is also an answer set in DI-semantics, but not conversely; i.e., DI-semantics is a relaxation of GL-semantics.

For the GSC problem in Example 1, every candidate answer set of Π in DI-semantics satisfies the first three conditions of a strategic set, so answer sets of Π in DI-semantics correspond to strategic sets for the GSC problem. As a result, the expected answer set $I = \{g_1, g_2, c_1, c_2\}$ of Π, corresponding to the strategic set $C' = \{c_1, c_2\}$, is an answer set in DI-semantics, though it is not an answer set in GL-semantics.

Next, we further use instances of the GSC problem as typical examples to show that the requirements of constraint monotonicity and foundedness may exclude expected answer sets for some simple disjunctive programs.

[2]As analyzed in [Shen and Eiter, 2019], the operator "|" amounts in the GL-semantics for simple disjunctive logic programs to the classical connective "$\vee$". However, [Gelfond and Lifschitz, 1991] deliberately used "|" to indicate that the connection of alternatives in the rule head is not simply disjunction "$\vee$" as in classical logic. Rather, one "commits" to an alternative A_i in a rule head $A_1 \mid \cdots \mid A_k$ for the belief state that represents the model one is willing to accept. In [Gelfond, 2008; Gelfond and Kah, 2014] "*or*" is used in place of "|" and called *epistemic disjunction*.

Example 2 (counter-example to constraint monotonicity)**.** *Consider again the instance of the GSC problem in Example 1. Observe that there will be no strategic set $C' \subseteq C$ satisfying the strategy condition σ_1^3. Assume on the contrary that σ_1^3 is true, i.e., company c_1 is in C' and c_2 is not, then (by condition (2) of a strategic set) c_3 must be in C'; this in turn makes σ_1^2 true and thus company c_2 must also be in C', a contradiction.*

Correspondingly, the program Π in Example 1 will have no answer set I satisfying the body condition $c_1 \wedge \neg c_2$ of rule (7). If so, i.e., c_1 is in I and c_2 is not, then by rules (7) and (6), c_3 and c_2 must also be in I, a contradiction.

The above observation shows that the strategy condition σ_1^3 and rule (7) play in view of condition σ_1^2 and rule (6) the role of a constraint for strategic set and answer set construction, respectively. That is, rule (7) amounts to a constraint

$$\bot \leftarrow c_1 \wedge \neg c_2 \qquad (7')$$

Let Π_1 be Π with rule (7) replaced by $(7')$. Then Π_1 and Π have the same expected answer set $I = \{g_1, g_2, c_1, c_2\}$, which corresponds to the strategic set $C' = \{c_1, c_2\}$ for the GSC problem.[3]

We now show that the strategic set $C' = \{c_1, c_2\}$ violates constraint monotonicity. One can easily check that without the strategy condition σ_1^3, this GSC problem has the only strategic set $D = \{c_1\}$. However, when σ_1^3 is added, which plays the role of a constraint like rule $(7')$ for strategic set construction, this GSC problem has the only strategic set $C' = \{c_1, c_2\}$. This violates constraint monotonicity that requires every strategic set of the GSC problem containing σ_1^3 should be a strategic set of the GSC problem without σ_1^3.

Correspondingly, we show that the expected answer set $I = \{g_1, g_2, c_1, c_2\}$ for Π_1 violates constraint monotonicity. Let Π' consist of rules (1) to (6) in Π_1. It is easy to check that $J = \{g_1, g_2, c_1\}$ is the only answer set of Π'. As Π_1 is Π' plus the constraint $(7')$, i.e., $\Pi_1 = \Pi' \cup \{(7')\}$, constraint monotonicity requires that answer sets of Π_1 should be answer sets of Π' satisfying the constraint $(7')$. As the only answer set J of Π' does not satisfy the constraint $(7')$, if we enforce constraint monotonicity, Π_1 would have no answer set. This reveals that constraint monotonicity may exclude expected answer sets for some simple disjunctive programs.

Note that $I = \{g_1, g_2, c_1, c_2\}$ is the only answer set of Π_1 in DI-semantics, corresponding to the strategic set $C' = \{c_1, c_2\}$; however, Π_1 has no answer set in GL-semantics.

Discussion: It is the "subset-minimal" condition that makes some strategic sets like $C' = \{c_1, c_2\}$ and answer sets like $I = \{g_1, g_2, c_1, c_2\}$ in the above example violate the property of constraint monotonicity. Without the constraint σ_1^3, C' satisfies the first three conditions of a strategic set, but it is not subset-minimal ($D = \{c_1\}$ is subset-minimal) and thus is not a strategic set. However, C' becomes subset-minimal when σ_1^3 is added. This explains why C' is a strategic set of the GSC problem containing σ_1^3, but it is not a strategic set when the constraint is removed. For the same reason, I

[3]We note that this replacement preserves moreover strong equivalence of Π [Lifschitz *et al.*, 2001] and thus should be regarded as inessential under the GL-semantics.

becomes subset-minimal and is an answer set of the simple disjunctive program Π_1 containing the constraint ($7'$), but it is not an answer set when the constraint is removed.

Example 3 (counter-example to foundedness)**.** *Next we show that the expected answer set $I = \{g_1, g_2, c_1, c_2\}$ of Π in Example 1, which corresponds to the strategic set $C' = \{c_1, c_2\}$ for the GSC problem, violates the property of foundedness, i.e., some subset X of I is an unfounded set (see the three conditions (f1)-(f3) of an unfounded set in the introduction).*

Consider $X = \{c_2\}$. The program Π in Example 1 has rules (3) and (6), i.e., $c_1 \mid c_2 \leftarrow g_1$ and $c_2 \leftarrow c_3$, whose heads intersect with X. For rule (3), as $(\{c_1, c_2\} \setminus X) \cap I = \{c_1\}$, condition (f3) is violated, while for rule (6), as I does not satisfy c_3, condition (f1) is violated. Consequently, no rule in Π whose head intersects with X satisfies the conditions (f1)-(f3); thus X is an unfounded set of I.

Therefore, the requirement of foundedness would exclude the expected answer set $I = \{g_1, g_2, c_1, c_2\}$ and the corresponding strategic set $C' = \{c_1, c_2\}$.

Discussion: It is the restriction of condition (f3), i.e., $(\{A_1, \cdots, A_k\} \setminus X) \cap I = \emptyset$, that makes the answer set $I = \{g_1, g_2, c_1, c_2\}$ resp. the strategic set $C' = \{c_1, c_2\}$ in Example 1 unfounded. The GSC problem says that g_1 can be produced either by c_1 or c_2, and g_2 produced either by c_1 or c_3. So we can choose c_2 from the head of rule (3) (i.e., we can assume that c_2 produces g_1) and c_1 from the head of rule (4) (i.e., c_1 produces g_2). This yields the answer set I resp. the strategic set C'. However, by condition (f3) once c_1 is selected from the head of rule (4), selecting c_2 from the head of rule (3) is disallowed; otherwise, $(\{c_1, c_2\} \setminus X) \cap I = \{c_1\} \neq \emptyset$ would hold, leading to an unfounded set $X = \{c_2\}$ of I.

4 Considering Constraint Monotonicity and Foundedness for Epistemic Specifications

In [Kahl and Leclerc, 2018], the notion of constraint monotonicity for simple disjunctive programs was extended to epistemic specifications. In particular, they introduced a notion of *subjective constraint monotonicity*, which requires that for any epistemic specification Π and subjective constraint C, world views of $\Pi \cup \{C\}$ should be world views of Π satisfying C. Here a subjective constraint is a constraint $\bot \leftarrow G_1 \wedge \cdots \wedge G_n$, where each G_i is a modal literal of the form $\mathbf{K}A$, $\neg\mathbf{K}A$, $\mathbf{M}A$ or $\neg\mathbf{M}A$, and A is an atom.

We also use an instance of the GSC problem as a typical example to show that subjective constraint monotonicity may exclude expected world views for epistemic specifications.

Example 4 (counter-example to subjective constraint monotonicity)**.** *Consider a slightly different instance of the GSC problem, which is obtained from the instance in Example 1 simply by replacing the strategy condition $\sigma_1^3 = c_1 \wedge \neg c_2$ with $\sigma_1^3 = \neg c_2$. This instance can naturally be encoded as a simple disjunctive program Π_2, which is Π in Example 1 with the last rule replaced by the rule*

$$c_3 \leftarrow \neg c_2 \qquad \text{// strategy condition } \sigma_1^3 = \neg c_2 \qquad (7)$$

It is easy to check that Π_2 has only one expected answer set $I = \{g_1, g_2, c_1, c_2\}$, which corresponds to the only strategic set $C' = \{c_1, c_2\}$ for this instance of the GSC problem.

Observe that this instance of the GSC problem will have no strategic set C' satisfying the strategy condition σ_1^3. Actually, σ_1^3 plays the role of a constraint requiring that every strategic set C' should contain company c_2. If σ_1^3 is true, i.e., some C' does not contain c_2, then (by condition (2) of a strategic set) c_3 must be in C'; this in turn makes σ_1^2 true and thus c_2 must also be in C', a contradiction.

As strategic sets of this instance of the GSC problem one-to-one correspond to answer sets of Π_2, the above observation suggests that Π_2 will have no answer set satisfying the body condition $\neg c_2$ of rule (7). That is, rule (7) in Π_2 for σ_1^3 amounts to a constraint

$$\bot \leftarrow \neg c_2 \qquad (7')$$

which requires that every answer set of Π_2 should contain c_2. Such a constraint on "every answer set" can also be encoded as a subjective constraint with a modal literal $\mathbf{K}c_2$:

$$\bot \leftarrow \neg\mathbf{K}c_2 \quad \text{// each strategic set/answer set contains } c_2 \; (7'')$$

Let Π_3 and Π_4 be Π_2 with rule (7) replaced by ($7'$) and ($7''$), respectively. Then arguably Π_2 and Π_3 should have the same answer sets,[4] *i.e., they have only one expected answer set $I = \{g_1, g_2, c_1, c_2\}$. Likewise, Π_3 and Π_4 should admit the same answer sets.*

Note that Π_4 is an epistemic specification, so finding the set $\mathcal{S}$ of all strategic sets for the GSC problem corresponds to constructing the world view $\mathcal{W}$ consisting of all expected answer sets of Π_4 corresponding to $\mathcal{S}$. As Π_4 has only one expected answer set $I = \{g_1, g_2, c_1, c_2\}$, it is expected that Π_4 has the only world view $\mathcal{W} = \{I\}$, which corresponds to the set $\mathcal{S} = \{C'\}$ of all strategic sets, where $C' = \{c_1, c_2\}$.

We next show that this expected world view $\mathcal{W} = \{\{g_1, g_2, c_1, c_2\}\}$ for Π_4 violates the property of subjective constraint monotonicity. Let Π' consist of rules (1)-(6) in Π_2. Then $J = \{g_1, g_2, c_1\}$ is the only answer set of Π' and thus $\mathcal{W}' = \{\{g_1, g_2, c_1\}\}$ is the only world view of Π'. As Π_4 is Π' plus the subjective constraint ($7''$), the subjective constraint monotonicity requires that world views of Π_4 should be world views of Π' satisfying the constraint ($7''$). As the only world view $\mathcal{W}'$ of Π' does not satisfy the constraint ($7''$), if we require the subjective constraint monotonicity, Π_4 would have no world view. This reveals that subjective constraint monotonicity may exclude expected world views for some epistemic specifications.

Another stronger requirement on world view semantics, called *epistemic splitting*, was introduced in [Cabalar *et al.*, 2019]. Informally, an epistemic specification can be split if its *top* part only refers to the atoms of the *bottom* part through modal literals. Then, a world view semantics is said to satisfy epistemic splitting if it is possible to get its world views by first obtaining the world views of the bottom and then using the modal literals in the top as *queries* on the bottom part previously obtained. It was shown in [Cabalar *et al.*, 2019] that the epistemic splitting property is even more restrictive than

[4] Note in particular that no modal literals occur in rules (1)-(6).

subjective constraint monotonicity in that every world view semantics satisfying epistemic splitting also satisfies subjective constraint monotonicity. Due to this, epistemic splitting may exclude some expected world views for epistemic specifications like that in Example 4.

Furthermore, in [Cabalar *et al.*, 2020] the notion of foundedness for simple disjunctive programs was extended to epistemic specifications. Following their convention, for a rule r in an epistemic specification, we denote with $Head(r)$ the set of atoms in the head of r and with $Body_{sub}(r)$ and $Body_{ob}(r)$ those atoms occurring in the body of r in modal and non-modal literals, respectively. Moreover, $Body^+_{sub}(r)$ (resp. $Body^+_{ob}(r)$) refers to those atoms in $Body_{sub}(r)$ (resp. $Body_{ob}(r)$) occurring in positive literals.

Definition 3 (unfounded set [Cabalar *et al.*, 2020]). *Let Π be an epistemic specification and $\mathcal{W}$ a world view. An* unfounded set *w.r.t. $\mathcal{W}$ is a set $S \neq \emptyset$ of pairs $\langle X, I\rangle$, where X and I are sets of atoms, such that for each $\langle X, I\rangle \in S$, no rule r in Π with $Head(r) \cap X \neq \emptyset$ satisfies (ef1)-(ef4):*

(ef1) I satisfies the body condition of r w.r.t. $\mathcal{W}$;
(ef2) $Body^+_{ob}(r) \cap X = \emptyset$;
(ef3) $(Head(r) \setminus X) \cap I = \emptyset$; and
(ef4) for all $\langle X', I'\rangle \in S$, $Body^+_{sub}(r) \cap X' = \emptyset$.

Note that condition (ef2) generalizes (f2) while (ef4) is new. Founded world views are then as follows.

Definition 4 (founded world view [Cabalar *et al.*, 2020]). *A world view $\mathcal{W}$ of an epistemic specification Π is* unfounded *if there is some unfounded set S s.t. every $\langle X, I\rangle \in S$ satisfies $X \cap I \neq \emptyset$ and $I \in \mathcal{W}$; otherwise, $\mathcal{W}$ is called* founded.

Again we use an instance of the GSC problem as an example to show that the foundedness property may exclude some expected world views for epistemic specifications.

Example 5 (counter-example to foundedness). *Consider again the epistemic specification Π_4 for the GSC problem in Example 4, where $\mathcal{W} = \{\{g_1, g_2, c_1, c_2\}\}$ is the expected world view of Π_4, which corresponds to the set $\{\{c_1, c_2\}\}$ of all strategic sets for the GSC problem.*

We show that $\mathcal{W}$ violates the property of foundedness; i.e., there is an unfounded set S w.r.t. $\mathcal{W}$ such that every $\langle X, I\rangle \in S$ satisfies $I \in \mathcal{W}$ and $X \cap I \neq \emptyset$. Let $S = \{\langle X, I\rangle\}$, where $X = \{c_2\}$ and $I = \{g_1, g_2, c_1, c_2\}$. We show that for $\langle X, I\rangle$ no rule $r \in \Pi_4$ with $Head(r) \cap \{c_2\} \neq \emptyset$ satisfies (ef1)-(ef4).

Π_4 has rules (3) and (6) whose heads intersect with X. For rule (3), as $(\{c_1, c_2\} \setminus X) \cap I = \{c_1\}$, condition (ef3) is violated; for rule (6), as I does not satisfy its body condition c_3, condition (ef1) is violated. Consequently, no rule in Π_4 whose head intersects with X satisfies (ef1)-(ef4); thus $S = \{\langle X, I\rangle\}$ is an unfounded set w.r.t. $\mathcal{W}$. As $X \cap I \neq \emptyset$ and $I \in \mathcal{W}$, $\mathcal{W}$ is hence unfounded.

Therefore, if taking the foundedness requirement on world views of Π_4, we would lose $\mathcal{W} = \{\{g_1, g_2, c_1, c_2\}\}$. This reveals that the foundedness property may exclude expected world views for some epistemic specifications.

5 Related Work

[Kahl and Leclerc, 2018], [Cabalar *et al.*, 2020] and [Cabalar *et al.*, 2019] defined the notions of subjective constraint monotonicity, foundedness and epistemic splitting for epistemic specifications, respectively. [Fandinno *et al.*, 2021] provided a survey and compared the existing world view semantics based on how they comply with the three properties.

[Shen and Eiter, 2020] observed that the aforementioned properties may be too strong in general and used some abstract programs to demonstrate it. [Su, 2021] said that the soundness of the three properties is still under debate, and [Costantini, 2021] proposed an alternative yet less restrictive version of the epistemic splitting property.

In this paper, following the observation of [Shen and Eiter, 2020] we further introduced instances of the GSC problem [Cadoli *et al.*, 1997; Leone *et al.*, 2006; Shen and Eiter, 2019] as concrete examples to demonstrate that the three properties may exclude some expected answer sets and world views. The programs in Examples 1 and 2 show this for simple disjunctive programs, where the GL-semantics does not single out the expected strategic set $C' = \{c_1, c_2\}$. Furthermore, the expected world view $\mathcal{W} = \{\{g_1, g_2, c_1, c_2\}\}$ for Π_4 in Examples 4 and 5, which corresponds to the collection $\mathcal{S} = \{\{c_1, c_2\}\}$ of all strategic sets for the GSC problem, violates subjective constraint monotonicity and foundedness. This world view can be obtained by the generic world view semantics of [Shen and Eiter, 2016, Definition 8], where the base answer set semantics $\mathcal{X}$ takes the DI-semantics. As far as we can determine, it is not obtained by other world view semantics such as those in [Gelfond, 1991; Kahl *et al.*, 2020; Kahl *et al.*, 2019; Su *et al.*, 2020; Cabalar *et al.*, 2020].

6 Conclusions

In ASP, a problem is encoded as a logic program whose answer sets correspond to solutions to the problem. The GSC problem for ASP benchmark competitions is a particularly useful example in that it can be naturally represented by a simple disjunctive program Π as in Example 1 such that the expected answer sets of Π by intuition about rule satisfaction and rationality in terms of knowledge minimization in line with the Gelfond's informal principles [Gelfond, 2008] correspond one-to-one to strategic sets of the GSC problem. In this paper we introduced different instances of the GSC problem whose expected answer sets and world views violate the properties of constraint monotonicity and foundedness. They may be viewed as counter-examples to making these properties mandatory requirements for an answer set and world view semantics, in particular with respect to modeling and when knowledge minimization should be exploited as an asset of nonmonotonic semantics. This provides an answer to the fundamental question that we raised in the introduction.

Now that constraint monotonicity and foundedness may be seen as non-mandatory properties for an answer set/world view semantics in general, a natural question is what else are (perhaps similar) properties that every answer set/world view semantics is expected to satisfy? This presents an interesting and challenging open question for ongoing study in ASP.

Acknowledgments

We thank the reviewers for their helpful comments. This work was supported in part by NSFC grant 61976205 and the EU grant HumanE-AI-Net (ICT-48-2020-RIA 952026).

References

[Cabalar *et al.*, 2019] Pedro Cabalar, Jorge Fandinno, and Luis Fariñas del Cerro. Splitting epistemic logic programs. In *Proc. LPNMR 2019*, pages 120–133, 2019.

[Cabalar *et al.*, 2020] Pedro Cabalar, Jorge Fandinno, and Luis Fariñas del Cerro. Autoepistemic answer set programming. *Artif. Intell.*, 289:103382, 2020.

[Cadoli *et al.*, 1997] Marco Cadoli, Thomas Eiter, and Georg Gottlob. Default logic as a query language. *IEEE TKDE*, 9(3):448–463, 1997.

[Costantini, 2021] Stefania Costantini. Top-down splitting property for epistemic logic programs. In *Proceedings of the International Conference on Logic Programming 2021 Workshops*, volume 2970 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2021.

[del Cerro *et al.*, 2015] Luis Fariñas del Cerro, Andreas Herzig, and Ezgi Iraz Su. Epistemic equilibrium logic. In *Proc. IJCAI 2015*, pages 2964–2970, 2015.

[Faber *et al.*, 2011] Wolfgang Faber, Gerald Pfeifer, and Nicola Leone. Semantics and complexity of recursive aggregates in answer set programming. *Artif. Intell.*, 175(1):278–298, 2011.

[Fandinno *et al.*, 2021] Jorge Fandinno, Wolfgang Faber, and Michael Gelfond. Thirty years of epistemic specifications. *CoRR*, abs/2108.07669, 2021.

[Ferraris *et al.*, 2011] Paolo Ferraris, Joohyung Lee, and Vladimir Lifschitz. Stable models and circumscription. *Artif. Intell.*, 175(1):236–263, 2011.

[Gelfond and Kah, 2014] Michael Gelfond and Yulia Kah. *Knowledge Representation, Reasoning, and the Design of Intelligent Agents*. Cambridge University Press, 2014.

[Gelfond and Lifschitz, 1988] Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In *Proc. ICLP 1988*, pages 1070–1080, 1988.

[Gelfond and Lifschitz, 1991] Michael Gelfond and Vladimir Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9:365–385, 1991.

[Gelfond, 1991] Michael Gelfond. Strong introspection. In *Proc. AAAI 1991*, pages 386–391, 1991.

[Gelfond, 2008] Michael Gelfond. Answer sets. In *Handbook of Knowledge Representation*, pages 285–316. 2008.

[Hecher *et al.*, 2020] Markus Hecher, Michael Morak, and Stefan Woltran. Structural decompositions of epistemic logic programs. In *Proc. AAAI 2020*, pages 2830–2837, 2020.

[Kahl and Leclerc, 2018] Patrick Thor Kahl and Anthony Leclerc. Epistemic logic programs with world view constraints. In *Proc. ICLP 2018*, pages 1–17, 2018.

[Kahl *et al.*, 2019] Patrick Thor Kahl, Anthony P. Leclerc, and Tran Cao Son. A parallel memory-efficient epistemic logic program solver: harder, better, faster. *Ann. Math. Artif. Intell.*, 86(1-3):61–85, 2019.

[Kahl *et al.*, 2020] Patrick Kahl, Richard Watson, Evgenii Balai, Michael Gelfond, and Yuanlin Zhang. The language of epistemic specifications (refined) including a prototype solver. *Journal of Logic and Computation*, 30(4):953–989, 2020.

[Kahl, 2014] Patrick Thor Kahl. *Refining the semantics for epistemic logic programs*. PhD thesis, Texas Tech Univ., USA, 2014.

[Leone *et al.*, 1997] Nicola Leone, Pasquale Rullo, and Francesco Scarcello. Disjunctive stable models: Unfounded sets, fixpoint semantics, and computation. *Information and Compututation*, 135(2):69–112, 1997.

[Leone *et al.*, 2006] Nicola Leone, Gerald Pfeifer, Wolfgang Faber, Thomas Eiter, Georg Gottlob, Simona Perri, and Francesco Scarcello. The dlv system for knowledge repsentation and reasoning. *ACM Trans. Comput. Logic*, 7(3):499–562, July 2006.

[Lifschitz *et al.*, 1999] Vladimir Lifschitz, Lappoon Tang, and Hudson Turner. Nested expressions in logic programs. *Ann. Math. Artif. Intell.*, 25(1-2):369–389, 1999.

[Lifschitz *et al.*, 2001] Vladimir Lifschitz, David Pearce, and Agustín Valverde. Strongly equivalent logic programs. *ACM Trans. on Computat. Logic*, 2(4):526–541, 2001.

[Lifschitz, 2019] Vladimir Lifschitz. *Answer Set Programming*. Springer, 2019.

[Pearce, 2006] David Pearce. Equilibrium logic. *Ann. Math. Artif. Intell.*, 47(1-2):3–41, 2006.

[Shen and Eiter, 2016] Yi-Dong Shen and Thomas Eiter. Evaluating epistemic negation in answer set programming. *Artif. Intell.*, 237:115–135, 2016.

[Shen and Eiter, 2019] Yi-Dong Shen and Thomas Eiter. Determining inference semantics for disjunctive logic programs. *Artif. Intell.*, 277:1–28, 2019.

[Shen and Eiter, 2020] Yi-Dong Shen and Thomas Eiter. Constraint monotonicity, epistemic splitting and foundedness are too strong in answer set programming. *CoRR*, abs/2010.00191, 2020.

[Shen *et al.*, 2014] Yi-Dong Shen, Kewen Wang, Thomas Eiter, Michael Fink, Christoph Redl, Thomas Krennwallner, and Jun Deng. FLP answer set semantics without circular justifications for general logic programs. *Artif. Intell.*, 213:1–41, 2014.

[Su *et al.*, 2020] Ezgi Iraz Su, Luis Fariñas del Cerro, and Andreas Herzig. Autoepistemic equilibrium logic and epistemic specifications. *Artif. Intell.*, 282:103–249, 2020.

[Su, 2021] Ezgi Iraz Su. Refining the semantics of epistemic specifications. In *Proc. ICLP 2021*, pages 113–126, 2021.

[Truszczynski, 2010] Miroslaw Truszczynski. Reducts of propositional theories, satisfiability relations, and generalizations of semantics of logic programs. *Artif. Intell.*, 174(16-17):1285–1306, 2010.