Towards a Unified Framework for Declarative Knowledge-Change—Principles and Consistency

Juan C. Acosta-Guadarrama*

Research Group of Computer Systems
Autonomous University of Mexico State,
FI-UAEM, Mexico
jguadarrama@gmail.com

Abstract. As suggested in the literature, revising and updating beliefs and knowledge bases is an important yet unsolved topic in knowledge representation and reasoning in Answer Set Programming (ASP) that requires a solid theoretical basis, particularly in current applications of Artificial Intelligence where an agent can work in an open dynamic environment with incomplete information. Various researchers have combined principles to incorporate new information, and ASP as key components to set up their approaches. However, many of such proposals still present quite a few disadvantages when dealing with persistence situations, redundant information, contradictions or they simply lack of *further analysis of properties* that should make them more accessible. In need to satisfy more general principles and define a common frame of reference, this paper introduces a general framework for updates of logic programs, its *characterisation* by relevant belief-change principles, and an analysis of consistency. Rather than a sequence of updates of programs, this semantics consists in performing updates of *epistemic states* at the object level that meets well-accepted *belief revision principles* and follows their original conception.

1 Introduction

One of the goals of Artificial Intelligence and in particular of commonsense reasoning is how to define an intelligent agent that can be autonomous and that can act in an open dynamic environment. As suggested in the logic-programming literature, such a goal requires a solid theoretical basis on knowledge representation and nonmonotonic reasoning, and in particular, in *knowledge change*. Logic programming is a classical well-known mechanism to encode and represent agents' knowledge by means of a set of clauses (rules) called logic program. Such a program might be called a knowledge base and we encode it into a semantics called Answer Sets Programming [12] or ASP from now on. Tipically, logic programming has been static, however, in the sense that it provides no mechanism to automatically make changes (say, revision or updates) to a knowledge base.

* This project has been supported by the Public-Education Ministry, SEP, and by The Mexican Council of Science and Technology, CONACYT.

In particular, when updating knowledge one needs a way to avoid inconsistencies due to *potential contradictory information* upcoming from new evidence that is typically incomplete. There are several works in the context of logic programming based on a common ASP basis by satisfying certain properties and postulates: [10, 6, 23, 9]. However, despite the existence of several semantics for updates [2] and a vast analysis of general properties [10, 20], we claim we are still far from having a general one that can satisfy many existing and well-known principles to represent *"correct" dynamic knowledge.*

For instance, one of the missing and obvious properties in current semantics for updates is *persistence*, which others do not manage well for several reasons, mainly for their approach on *sequences of updates* [2]. Such a problem has been introduced and overcome by [22] by means of a semantics based on an extended particular version of abductive logic programming [13]. The semantics, however, is strongly based on syntactical changes that can be a problem (described later) and lacks of a proper characterisation of principles for updates (or belief revision), presumably due to their different goals of their referred approach.

With the aim to define a *general characterised semantics* and to succeed in the mentioned persistence situation (and in many others more), this paper series proposes an alternate more-general approach, founded on generalised answer sets, and based upon well-known principles for belief change (AGM-postulates [5]) that make it *syntax independent*, more intuitive and have generally-accepted structural properties.

Besides satisfying nearly all belief-revision postulates, the mathematical framework hereby proposed, consists in performing *successive updates* to a given epistemic state so as to deal with the problem that, according to [22], produces counterintuitive interpretations in most approaches.

Partial results of this article have already appeared in preliminary versions (without proofs or other properties) in [1], as well as in the extended abstract in [3, 4].

2 Preliminaries

A main foundation of this proposal are the well-known AGM-postulates [5] in a particular interpretation and notation, followed by a brief basic background of Answer Sets and Generalised Answer Sets. Intuitionistic logic, Nelson's logic and $\mathcal{N}_2$ are some of the logical systems that characterise ASP, and they represent evidence of the solid theoretical background to this work. Due to space constraints. however, the paper includes no sections to such strong foundation, which is easily accessible from the literature. Finally, to read this paper it is assumed that the reader is familiar with basic notions of AGM-theory, as well as logic programming and in particular with ASP.

2.1 Belief-change

One of the first, most important and now classical contributions to define properties of belief change is due to [5] for their eight *principles that a belief-revision* *function should satisfy*. They defined three types of operations: *contraction*, *revision* and *expansion*. The focus of this paper, however, is just on the two latter, hereby denoted as "$\circ$" and "$\uplus$", respectively.

After the original AGM-formulation, there have been several particular redefinitions and interpretations, which can be explained and justified in another version of this paper. The postulates version this research work has been proposing since [3] is [8]'s, in my "particular" interpretation. In our own notation, [8]'s redefinition of the original AGM-postulates corresponds to $(\mathsf{R} \circ 1)$–$(\mathsf{R} \circ 6)$, which I call KM′-postulates, where the general binary update operator "$\circ$" performs an update χ over an epistemic state $\mathcal{E}$ as formally shown below:

$(\mathsf{R} \circ 1)$ $Bel(\mathcal{E} \circ \chi)$ implies χ.
$(\mathsf{R} \circ 2)$ If $Bel(\mathcal{E}) \wedge \chi$ is satisfiable, then $Bel(\mathcal{E} \circ \chi) \equiv Bel(\mathcal{E}) \wedge \chi$.
$(\mathsf{R} \circ 3)$ If χ is satisfiable, then $Bel(\mathcal{E} \circ \chi)$ is also satisfiable.
$(\mathsf{R} \circ 4)$ If $\mathcal{E}_1 = \mathcal{E}_2$ and $\chi_1 \equiv \chi_2$ then $Bel(\mathcal{E}_1 \circ \chi_1) \equiv Bel(\mathcal{E}_2 \circ \chi_2)$.
$(\mathsf{R} \circ 5)$ $Bel(\mathcal{E} \circ \chi) \wedge \mu$ implies $Bel(\mathcal{E} \circ (\chi \wedge \mu))$.
$(\mathsf{R} \circ 6)$ If $Bel(\mathcal{E} \circ \chi) \wedge \mu$ is satisfiable, then $Bel(\mathcal{E} \circ (\chi \wedge \mu))$ implies $Bel(\mathcal{E} \circ \chi) \wedge \mu$.

where the general binary revision operator "$\circ$" performs over an *epistemic state* $\mathcal{E}$ updated with a *propositional sentence* χ, which results in a new epistemic state[1]. That is to say, $\mathcal{E} \circ \chi$ is an epistemic state.

Further explanation on this redefinition and justification is matter for a different version of this paper. By now, let us take these postulates for granted —i.e. unquestioned.

2.2 Logic Programming and Answer Sets

The following gives the description of ASP, which is identified with other names like *Stable Logic Programming* or *Stable Model Semantics* [12] and A-Prolog. Its formal language and some more notation are introduced from the literature as follows.

Definition 1 (ASP Language of logic programs, $\mathcal{L}_{\mathsf{ASP}}$). *In the following $\mathcal{L}_{ASP}$ is a language of propositional logic with propositional symbols: $a_0, a_1, \ldots$; connectives: "," (conjunction) and meta-connective ";"; disjunction "$\vee$", also denoted as "|"; $\leftarrow$ (derivation, also denoted as $\rightarrow$); propositional constants $\bot$ (falsum); $\top$ (verum); "$\neg$" (default negation or weak negation, also denoted with the word* not*); "$\sim$" (strong negation, equally denoted as "$-$"); auxiliary symbols: "(", ")" (parentheses). The propositional symbols are also called atoms or atomic propositions. A* literal *is an atom or a strong-negated atom. A* rule *ρ is an ordered pair $\mathsf{Head}(\rho) \leftarrow \mathsf{Body}(\rho)$, where $\mathsf{Head}(\rho)$ is a set of literals in disjunction and $\mathsf{Body}(\rho)$ a set of literals in conjunction.*

With the notation introduced in Definition 1, one may construct clauses of the following general form that are well known in the literature.

[1] Informally, they call it so for both representing a knowledge base and its interpretation.

Definition 2 (EDLP). *An* extended disjunctive logic program *is a set of rules of form*

$$\ell_1 \vee \ell_2 \vee \ldots \vee \ell_l \leftarrow \ell_{l+1}, \ldots, \ell_m, \neg\ell_{m+1}, \ldots, \neg\ell_n \quad (1)$$

where ℓ_i is a literal and $0 \leq l \leq m \leq n$.

In the individual case of this paper, we employ extended logic programs (ELP) as a particular instance of an EDLP, which permits no disjunction in the head of a rule (1).

2.3 Equivalence in Logic Programming

There are several kinds of equivalence in the literature, particularly in ASP and *monotonic logics* [19, 21, 14, 11]. Since ASP programs may be expressed in some monotonic logics, one may take advantage of checking equivalence in either system. In this paper I use $\mathcal{N}_2$-logic as one of its fundamental basis that characterises ASP, as well as a translation function between programs and $\mathcal{N}_2$ theories.

When establishing a relation between $\mathcal{N}_2$ and ASP, a *translation function* between ASP knowledge bases and $\mathcal{N}_2$ theories is necessary. The function is similar to the one from [19]:

Definition 3 (19). *The mapping function $T_{\mathcal{N}_2}(\cdot)$ translates an EDLP into propositional formulas of Nelson's logic $\mathcal{N}_2$.*

The rule $p_0 \vee p_1 \vee \cdots \vee p_l \leftarrow q_1, \ldots, q_m, \neg q_{m+1}, \ldots, \neg q_n$ is mapped into the formula $(q_1 \wedge \cdots \wedge q_m \wedge \neg q_{m+1} \wedge \cdots \wedge \neg q_n) \supset p_0 \vee p_1 \vee \cdots \vee p_l$ and the strong-negation *propositional symbol "$\sim$" has the same meaning of the logical symbol "$\sim$" in $\mathcal{N}_2$.*

With this translation, one may redefine ASP in terms of $\mathcal{N}_2$-logic, which shall be useful to provide even more properties, discussed along this section. Further explanation on how the results of such a characterisation arose are matter for another version of this paper.

The main result of such characterisations that is more relevant to this paper, though, may be expressed by Theorem 1, by using the notation from [20].

To begin with, the notation $\tau \Vdash_{\mathcal{N}_2} \mathcal{M}$ is a shorthand for both τ is *consistent* and *derives* $\mathcal{M}$ in $\mathcal{N}_2$-logic.

Theorem 1 ([20]). *Let Ψ be a program over a set of atoms $\mathcal{A}$ and $\mathcal{M} \subseteq \mathfrak{L}_{\mathcal{A}}$ a consistent set of literals. The set $\mathcal{M}$ is an answer set of Ψ if and only if $T_{\mathcal{N}_2}(\Psi) \cup \neg\overline{\mathcal{M}} \cup \neg\neg\mathcal{M} \Vdash_{\mathcal{N}_2} \mathcal{M}$.*

It is worth noting that the two negations introduced in this paper have the same intuitive meaning in $\mathcal{N}_2$ and that $\neg\neg\mathcal{M} \leftrightarrow \mathcal{M}$ is not a theorem in such logic. With Definition 3, one may easily establish an equivalence relation between ASP programs for some upcoming update properties:

Theorem 2 ([19]). *For any programs Ψ_1 and Ψ_2, $T_{\mathcal{N}_2}(\Psi_1) \equiv_{\mathcal{N}_2} T_{\mathcal{N}_2}(\Psi_2)$ if and only if for every program Ψ, $\Psi \cup \Psi_1$ and $\Psi \cup \Psi_2$ have the same Answer Sets.*

In order to simplify notation and with a slight abuse of notation, for any ASP programs Ψ_0, Ψ_1, $\Psi_0 \equiv_{\mathcal{N}_2} \Psi_1$ shall actually stand for $T_{\mathcal{N}_2}(\Psi_0) \equiv_{\mathcal{N}_2} T_{\mathcal{N}_2}(\Psi_1)$.

2.4 Abductive Programs and MGAS

As one of the semantics to interpret abductive programs, *Minimal Generalised Answer Sets* (MGAS) provides a more general and flexible semantics than standard ASP, with a wide range of applications. This framework is briefly introduced in the following set of definitions.

Definition 4 ([15]). *An* abductive logic program *is a pair* $\langle \Psi, \mathcal{A}^* \rangle$ *where* Ψ *is an arbitrary program and* $\mathcal{A}^*$ *a set of literals, called abducibles.*

On the other hand, there already exists a semantics to interpret abductive programs, called *generalised answer sets* (GAS) due to [15].

Definition 5 (GAS, [15]). *The expression* $\mathcal{M}(\Delta)$ *is a* generalised answer set *of the abductive program* $\langle \Psi, \mathcal{A}^* \rangle$ *if and only if* $\Delta \subseteq \mathcal{A}^*$ *and* $\mathcal{M}(\Delta)$ *is an answer set of* $\Psi \cup \{\alpha \leftarrow \top \mid \alpha \in \Delta\}$.

In case there are more than one generalised answer sets, an inclusion order may be established:

Definition 6 ([7]). *Let* $\mathcal{M}(\Delta_1)$ *and* $\mathcal{M}(\Delta_2)$ *be generalised answer sets of* $\langle \Psi, \mathcal{A}^* \rangle$*. The relation* $\mathcal{M}(\Delta_1) \leq_{\mathcal{A}^*} \mathcal{M}(\Delta_2)$ *holds if and only if* $\Delta_1 \subseteq \Delta_2$.

Last, one can easily establish the minimal generalised answer sets from an abductive inclusion order with the following definition.

Definition 7 (MGAS, [7]). *Let* $\mathcal{M}(\Delta)$ *be a* minimal generalised answer set *(MGAS) of* $\langle \Psi, \mathcal{A}^* \rangle$ *if and only if* $\mathcal{M}(\Delta)$ *is a generalised answer set of* $\langle \Psi, \mathcal{A}^* \rangle$ *and it is minimal with respect to its* abductive inclusion order.

3 Updating Epistemic States

One of the main goals of this proposal is satisfaction of most well-accepted *principles* for updates at the *object level* and in Minimal Generalised Answer Sets (MGAS), besides other basic properties. The approach consists in choosing the right interpretations for the desired properties, in an *iterated fashion*, rather than a sequence of updates like in earlier approaches.

Despite the nice advantages over other approaches, this semantics had not been characterised with more general principles, which is necessary both to avoid counterintuitive behavuour and to provide a common *frame of reference* to compare with other alternatives, which is one of the contributions of this work. So, let us briefly introduce it, followed by a *characterisation of Belief Revision.* The semantics can be formally introduced with the following set of definitions.

Formally, an *α-relaxed rule* is a rule ρ that is weakened by a given default-negated atom α in its body: $\mathsf{Head}(\rho) \leftarrow \mathsf{Body}(\rho) \cup \{\neg\alpha\}$. In addition, an *$\alpha$-relaxed program* is a set of α-relaxed rules. A *generalised program* of $\mathcal{A}^*$ is a set of rules of form $\{\ell \leftarrow \top \mid \ell \in \mathcal{A}^*\}$, where $\mathcal{A}^*$ is a given set of literals.

Accordingly, updating an extended logic program Ψ_1 with another Ψ_2 consists in transforming an ordered pair of programs into a single abductive program by means of a particular binary update operator "$\bullet$", as follows.

Definition 8 ($\bullet$-update Program, [1]). *Given an update ordered-pair $(\Psi_1 \bullet \Psi_2)$, of extended logic programs Ψ_1, Ψ_2, over a set of atoms $\mathcal{A}$; and a set of new distinguished abducibles $\mathcal{A}^*$, such that $\mathcal{A} \cap \mathcal{A}^* = \emptyset$; and the α-relaxed program Ψ' from Ψ_1, such that $\alpha \in \mathcal{A}^*$; and the abductive program $\Psi_{\mathcal{A}^*} = \langle \Psi' \cup \Psi_2, \mathcal{A}^* \rangle$. Its* $\bullet$-update program *is $\Psi' \cup \Psi_2 \cup \Psi_G$, where Ψ_G is a generalised program of $\mathcal{M} \cap \mathcal{A}^*$ for some* minimal generalised answer set *$\mathcal{M}$ of $\Psi_{\mathcal{A}^*}$ and "$\bullet$" is the corresponding update operator.*

Obviously Definition 8 allows none or more $\bullet$*-update programs*. In addition to that, Corollary 1 below shows that the update is always consistent provided that Ψ_1 is also consistent. Before that, let us formalise another minor obvious property:

Observation 31 *Let Ψ_G be a generalised program out of a minimal generalised answer set $\mathcal{M}$ from $\Psi_{\mathcal{A}^*}$ and $\mathcal{M}_1$ an answer set of Ψ_G. The following two statements hold:*

a) $\mathcal{M}_1 = \mathcal{M} \cap \mathcal{A}^*$.
b) $\mathcal{M}_1 \subseteq \mathcal{M}$.

Last but not least, a *model $\mathcal{S}$ of the new knowledge base* corresponds to an answer sets of a $\bullet$-update program as follows.

Definition 9 ($\bullet$-update Answer Set, [1]). *Let $\mathbf{\Psi}_\bullet = (\Psi_1 \bullet \Psi_2)$ be an update pair of extended logic programs over a set of atoms $\mathcal{A}$. Then, $\mathcal{S} \subseteq \mathcal{A}$ is a* $\bullet$-answer set *of $\mathbf{\Psi}_\bullet$ if and only if $\mathcal{S} = \mathcal{S}' \cap \mathcal{A}$ for some* minimal generalised answer set *$\mathcal{S}'$ of its $\bullet$-update program.*

Intuitively, this formulation establishes an order with respect to the *latest update* —which corresponds to [16, 17] first postulate $(\mathsf{R} \circ 1)$— and with respect to a *minimal change* when choosing the most preferred model: MGAS.

4 Properties

The following sets of properties of this simpler formulation are part the main contribution of this current semantics for *iterated updates* of *epistemic states*. They are classified into a study of consistency issues, and the satisfaction itself of KM'-postulates.

There are two particular properties suggested much earlier in [23] that are necessary for the rest of them. Additionally, the reader should note that a statement like $\Psi_1 \equiv \Psi_2$ means that both Ψ_1 and Ψ_2 have the same answer sets —or alternately $\Psi_1 \equiv_{\mathsf{ASP}} \Psi_2$. By a slight abuse of notation, when establishing equivalence between updates, indeed it means that they have the same (or different) update answer sets. Finally, the two properties from the literature (ref. [18, 20]), interpreted in our own notation, are the following.

- $\bullet$-SP-8, Strong Consistency, SC: If $\Psi_1 \cup \Psi_2$ is *consistent*, then $\Psi_1 \bullet \Psi_2 \equiv \Psi_1 \cup \Psi_2$. The update coincides with the union when $\Psi_1 \cup \Psi_2$ is consistent.
- $\bullet$-SP-9, Weak Irrelevance of Syntax, WIS: Let Ψ, Ψ_1, and Ψ_2 be logic programs under the same language. If $T_{\mathcal{N}_2}(\Psi_1) \equiv_{\mathcal{N}_2} T_{\mathcal{N}_2}(\Psi_2)$ then $\Psi \bullet \Psi_1 \equiv \Psi \bullet \Psi_2$.

Theorem 3. *[[1]] Suppose that Ψ, Ψ_1, Ψ_2 and Ψ_3 are ELP. Operator $\bullet$ satisfies the properties $\bullet$-SP-8 and $\bullet$-SP-9.*

4.1 Dealing with Inconsistencies

As previously suggested, dealing with inconsistencies is something necessary not only when new information contradicts previous one, but also with an *originally-inconsistent knowledge base*. This section consists of a study of consistency-preservation and consistency-restoration as key properties of $\bullet$-operator.

In particular, *Weak-consistency View* guarantees consistency of the abductive program from an update pair. On the other hand, a *consistent abductive program* from a $\bullet$-update pair shall mean the abductive program with generalised answer sets.

Lemma 1 (Weak Consistency View). *Suppose Ψ_0 and Ψ_1 are ELP's and an updating pair $(\Psi_0 \bullet \Psi_1)$ with its corresponding abductive program $\Psi_{\mathcal{A}^*} = \langle \Psi' \cup \Psi_1, \mathcal{A}^* \rangle$ over a set of atoms $\mathcal{A}$ and a set of new distinguished abducibles $\mathcal{A}^*$, such that $\mathcal{A} \cap \mathcal{A}^* = \emptyset$. If Ψ_1 is consistent then $\Psi_{\mathcal{A}^*}$ is also consistent.*

Proof. Suppose Ψ_0 and Ψ_1 are ELP's and that Ψ_1 is consistent. This means that the abductive program $\langle \Psi' \cup \Psi_1, \mathcal{A}^* \rangle$ is consistent and implies a generalised answer set $\mathcal{M}(\Delta)$ out of the answer sets of $\Psi' \cup \Psi_1 \cup \{\alpha \leftarrow \top \mid \alpha \in \Delta\}$, which is always consistent. Therefore, if Ψ_1 is consistent, then $\langle \Psi' \cup \Psi_1, \mathcal{A}^* \rangle$ is also consistent.

Accordingly, the following result holds.

Corollary 1 (Consistency Preservation). *Suppose Ψ_0 and Ψ_1 are ELP's. The update $\Psi_0 \bullet \Psi_1$ is consistent if Ψ_1 is consistent.*

Proof. The proof is similar to the one in Lemma 1

This property is known in the literature as *Consistency Preservation* and by [22] as *Inconsistency Removal.* Note that the sole name of the latter confirms the *syntactical orientation* of their approach. Last, it's worth noticing that this property is equivalent to postulate $(\mathsf{R} \circ 3)$ both [18]'s and [8]'s third postulate.

On the other hand, Ψ_1 inconsistent in Corollary 1 may lead to two possible situations: that the resulting update is either consistent or inconsistent, as shown in the following example.

Example 1 (Inconsistent Update). Suppose the update $\Psi_1 = \{a \leftarrow \neg a\}$, which has no answer sets, to an original *empty knowledge base* $\Psi_0 = \emptyset$. As a result, the update $\Psi_0 \bullet \Psi_1$ has no $\bullet$-update answer sets. Now suppose the same update to an initial knowledge base $\Psi_0' = \{a \leftarrow \top\}$. The $\bullet$-update answer set of such an update $\{a\} \models \Psi_0' \bullet \Psi_1$.

Corollary 1 also proves to be useful both when satisfying belief revision postulates and when *restoring consistency* from an originally inconsistent knowledge base, as explained below. On top of that, this property is a general case of [22]'s *inconsistency removal* that makes syntactical changes to the original knowledge base.

Next, the following proposition follows directly from Corollary 1.

Proposition 1 (Consistency Restoration). *Suppose Ψ_0 is an ELP. The update $\Psi_0 \bullet \emptyset$ is consistent.*

Proof. The proof is similar to the one in Lemma 1

As described in this section, $\bullet$-operator guarantees *robustness of knowledge bases* in many situations where other alternate frameworks break down. Accordingly, the properties presented in this section shall be part of a more general framework of principles and postulates.

Next, the core of this paper is the introduction of a particular interpretation of one of the latest formulations of the AGM-postulates (KM') and which of them are met by $\bullet$-operator.

4.2 Principles

One of the main goals of this paper is a formulation of a semantics for updates of logic programs that can meet as-many-as possible general properties. So, let us start this section with an interpretation and characterisation of [18]'s postulates $(\mathsf{R} \circ 1)$–$(\mathsf{R} \circ 6)$, over ELP's Ψ_i as a main foundation to this *revision* of *epistemic states*. Note that rather than "$\bullet$" operator, this set of postulates uses the general one "$\circ$" not to refer to any particular update semantics.

$(\mathsf{RG} \circ 1)$ $\Psi_1 \subseteq \Psi \circ \Psi_1$.
$(\mathsf{RG} \circ 2)$ If $\Psi \cup \Psi_1$ is consistent, then $\Psi \circ \Psi_1 \equiv_{\mathsf{ASP}} \Psi \cup \Psi_1$.
$(\mathsf{RG} \circ 3)$ If Ψ_1 is consistent, then $\Psi \circ \Psi_1$ is also consistent.
$(\mathsf{RG} \circ 4)$ If $\Psi_x = \Psi_y$ and $\Psi_1 \equiv_{\mathcal{N}_2} \Psi_2$ then $\Psi_x \circ \Psi_1 \equiv_{\mathsf{ASP}} \Psi_y \circ \Psi_2$.
$(\mathsf{RG} \circ 4')$ If $\Psi_1 \equiv_{\mathcal{N}_2} \Psi_2$ then $\Psi \circ \Psi_1 \equiv_{\mathsf{ASP}} \Psi \circ \Psi_2$.
$(\mathsf{RG} \circ 5)$ $\Psi \circ (\Psi_1 \cup \Psi_2) \subseteq (\Psi \circ \Psi_1) \cup \Psi_2$.

(RG ∘ 6) If $(\Psi \circ \Psi_1) \cup \Psi_2$ is consistent, then $(\Psi \circ \Psi_1) \cup \Psi_2 \subseteq \Psi \circ (\Psi_1 \cup \Psi_2)$.

An immediate result is the following main theorem of this paper certifying that $\bullet$-operator satisfies five of these six belief revision postulates.

Theorem 4 (RG ∘ -properties). *Suppose that Ψ, Ψ_1 and Ψ_2 are ELP. Update operator "$\bullet$" satisfies properties (RG ∘ 1)–(RG ∘ 4) and (RG ∘ 6).*

Proof. (RG ∘ 1) $\Psi_1 \subseteq \Psi \bullet \Psi_1$.

By Definition 8, $\Psi_1 \subseteq \Psi' \cup \Psi_G \cup \Psi_1$ that clearly satisfies the objective.

(RG ∘ 2) If $\Psi \cup \Psi_1$ is consistent, then $\Psi \bullet \Psi_1 \equiv \Psi \cup \Psi_1$.

This postulate corresponds to Strong-Consistency property and satisfied by Theorem 3.

(RG ∘ 3) If Ψ_1 is consistent, then $\Psi \bullet \Psi_1$ is also consistent.

This postulate is equivalent to Corollary 1.

(RG ∘ 4′) If $\Psi_1 \equiv_{\mathcal{N}_2} \Psi_2$ then $\Psi \bullet \Psi_1 \equiv \Psi \bullet \Psi_2$.

This postulate is equivalent to property $\bullet$-SP-9 and satisfied by Theorem 3.

(RG ∘ 6) If $(\Psi \bullet \Psi_1) \cup \Psi_2$ is consistent, then $(\Psi \bullet \Psi_1) \cup \Psi_2 \subseteq \Psi \bullet (\Psi_1 \cup \Psi_2)$.

Suppose $(\Psi \bullet \Psi_1) \cup \Psi_2$ is consistent. Then, the abductive program of $\Psi \bullet \Psi_1$ is $\langle \Psi' \cup \Psi_1, \mathcal{A}^* \rangle$ with its respective MGAS's $\mathcal{M}(\Delta_1)$ that is an answer set of $\Psi' \cup \Psi_1 \cup \{\alpha \leftarrow \top | \alpha \in \Delta_1\}$ where $\Delta_1 \subseteq \mathcal{A}^*$ and its corresponding generalised program Ψ_{G_1}. By Definition 8, the update $\Psi \bullet (\Psi_1 \cup \Psi_2)$ has the abductive program $\langle \Psi' \cup (\Psi_1 \cup \Psi_2), \mathcal{A}^* \rangle$ with its MGAS's $\mathcal{M}(\Delta_2)$ that is an answer set of $\Psi' \cup \Psi_1 \cup \Psi_2 \cup \{\alpha \leftarrow \top \mid \alpha \in \Delta_2\}$ where $\Delta_2 \subseteq \mathcal{A}^*$ and its corresponding generalised program Ψ_{G_2}. Because Ψ_2 is consistent with the update $\Psi \bullet \Psi_1$, the number of abducibles in Δ_1 never change, and it's easy to verify that Δ_2 contains at least the same abducibles than Δ_1, $\Delta_1 \subseteq \Delta_2 \subseteq \mathcal{A}^*$ and thus $\Psi_{G_1} \subseteq \Psi_{G_2}$. In consequence, the respective $\bullet$-update programs of each pair are $\Psi' \cup \Psi_1 \cup \Psi_2 \cup \Psi_{G_1}$ and $\Psi' \cup \Psi_1 \cup \Psi_2 \cup \Psi_{G_2}$, where $\Psi' \cup \Psi_1 \cup \Psi_2 \cup \Psi_{G_1} \subseteq \Psi' \cup \Psi_1 \cup \Psi_2 \cup \Psi_{G_2}$ as required. Therefore, $(\Psi \bullet \Psi_1) \cup \Psi_2 \subseteq \Psi \bullet (\Psi_1 \cup \Psi_2)$.

Nevertheless, postulate (RG ∘ 5) does not hold. As a counterexample, consider the following programs: $\Psi = \{a \leftarrow \top; {\sim}b \leftarrow \top; {\sim}c \leftarrow \top\}; \Psi_1 = \{b \leftarrow \top\}; \Psi_2 = \{c \leftarrow \top\}$. Such an update inverts the direction of the relation.

4.3 Discussion

This section is an introduction to general properties characterising $\bullet$-operator that go from the structural properties, most of them inherited from its equivalent counterpart in [23], to more general ones encoded in our particular interpretation of belief revision postulates. The satisfaction of AGM-postulates in ASP is something very important, provided that other current approaches either do not meet most of them or have discarded them for considering that their *monotonic nature* is incompatible with non-monotonic frameworks like ASP.

Another issue other approaches have is when updating in a sequence rather than an iterated fashion, which leads to counterintuitive results, especially in the persistence situation and when new updates arise afterwards. By following the original AGM paradigm, we also claim that the iterative approach has other more natural properties than its sequenced counterpart.

The section is also a *study of inconsistencies* not only due to new information that contradicts current knowledge, but also from both an *originally-inconsistent knowledge base*, as well as *new originally-inconsistent observations* that not necessarily contradict current beliefs. The former is something that may be considered a key feature of belief revision. However, as one of the main goals of this paper is to provide a strong general framework to correctly represent knowledge, and making a strict distinction with belief update theory might result controversial.

On the other hand, dealing with *originally-inconsistent observations* might seem counterintuitive to some researches, but it does not mean that observing such contradictions may not be possible in a *changing environment*. Take for example two *concurrent observations* that contradict each other, updating a current knowledge base in, say, a problem of Ambient Intelligence when a sensor fails and another one contradicts it. Another example is an *observation that is inconsistent* due to a typo or another kind of human error. Traditionally, those problems are left to future debugging, but with a tendency to model even-more *autonomous entities*, tolerating inconsistencies is not only reasonable, but also necessary to preserve a knowledge base from collapse.

5 Conclusions and Related Work

This paper consists of a *generalisation* of •-operator that satisfies five out of six suitable *belief-revision postulates* for *updating epistemic states*, as well as other useful properties from previous proposals, towards a more general formulation to update logic programs. Additionally, this paper has introduced a study of *consistency preservation* and *consistency recovery* (also known as *consistency restoration*) as a very important issue to be considered when updating logic programs that may help achieve particular needs and preserve the epistemic state from collapse. As a result, this mathematical framework provides a *strong theoretical foundation* on well-known principles and other fundamental properties shared mainly with other operators. Although •-operator overcomes the persistency problems that [22] have pointed out, because of its *object-level approach*, both proposals have lacked of a general and fundamental *characterisation of Belief-revision* postulates, and is here provided as a *frame of reference* and *comparison*. By combining the operational features of ASP, its characterising logics and nonmonotonic theory that underpin it, and a broad set of well-known belief-change principles, one should be capable of configuring agents' knowledge bases that are well-behaved and robust-enough against unexpected circumstances. In addition to that, there is an ongoing work from [9] who also pose our same hypothesis of the need to satisfy most belief revision postulates and indeed they claim to do so. However, they still lack of further comparison with more recent works on belief revision in ASP (their references on ASP-"update" operators are a bit too old and they discard new later approaches that overcome the pointed-out problems, for example) and thus they fail to make a strong justification for their alternate approach.

Bibliography

[1] Juan C. Acosta-G. Maintaining knowledge bases at the object level. In Alexander Gelbukh and Angel F. Kuri Morales, editors, *Special Session of the 6th International MICAI Conference*, pages 3–13, Aguascalientes, Mexico, November 2007. IEEE Computer Society. ISBN: 978-0-7695-3124-3.

[2] Juan C. Acosta-G. A road map of updating in ASP. ISSN: 1860-8477 IfI-07-16, Institute für Informatik, TU-Clausthal, Clausthal, Germany, December 2007. 35pp.

[3] Juan C. Acosta-G. AGM postulates in answer sets. In Mauricio Osorio and Iván Olmos, editors, *LANMR'08 Fourth Latin American Workshop on Non-monotonic Reasoning*, volume 408 of *ISSN 1613-0073*, page 3pp, Benemérita Universidad Autónoma de Puebla, Puebla, México, 2008. CEUR.

[4] Juan C. Acosta-G. and Jorge Hernández. A generalised semantics for belief updates—an equivalence-based approach. *Research in Computing Science*, 53:137–148, 2011.

[5] Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change: Partial meet contraction and revision functions. *The Journal of Symbolic Logic*, 50(2):510–530, June 1985.

[6] José Júlio Alferes, Federico Banti, Antonio Brogi, and João Alexandre Leite. The refined extension principle for semantics of dynamic logic programming. *Studia Logica*, 79(1):7–32, 2005.

[7] Marcello Balduccini and Michael Gelfond. Logic programs with consistency-restoring rules. In *Proceedings of the AAAI Spring 2003 Symposium*, pages 9–18, Palo Alto, California, 2003. AAAI Press.

[8] Adnan Darwiche and Judea Pearl. On the logic of iterated belief revision. In Ronald Fagin, editor, *Proceedings of the fifth Conference on Theoretical Aspects of Reasoning about Knowledge*, pages 5–23. Morgan Kaufmann, Pacific Grove, CA, 1994.

[9] James P. Delgrande, Torsten Schaub, Hans Tompits, and Stefan Woltran. Belief revision of logic programs under answer set semantics. In *KR*, pages 411–421, 2008.

[10] Thomas Eiter, Michael Fink, Giuliana Sabbatini, and Hans Tompits. On properties of update sequences based on causal rejection. *Theory and Practice of Logic Programming*, 2(6):711–767, 2002.

[11] Thomas Eiter, Michael Fink, Giuliana Sabbatini, and Hans Tompits. Reasoning about evolving nonmonotonic knowledge bases. *ACM Transactions on Computational Logic*, 6(2):389–440, 2005.

[12] Michael Gelfond and Vladimir Lifschitz. The Stable Model Semantics for Logic Programming. In Robert A. Kowalski and Kenneth A. Bowen, editors,

Logic Programming, Proceedings of the Fifth International Conference and Symposium ICLP/SLP, pages 1070–1080, Seattle, Washington, 1988. MIT Press.

[13] Katsumi Inoue and Chiaki Sakama. Abductive framework for nonmonotonic theory change. In *the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 204–210, Montreal, Canada, 1995. Morgan Kaufmann Publishers.

[14] Katsumi Inoue and Chiaki Sakama. Equivalence of logic programs under updates. In José Júlio Alferes and João Alexandre Leite, editors, *Proceedings of the 9th European Conference on Logics in Artificial Intelligence (JELIA'04)*, volume 3229 of *LNAI*, pages 174–186, Lisbon, Portugal, 2004. Springer-Verlag.

[15] Antonis C. Kakas and Paolo Mancarella. Generalized Stable Models: A semantics for abduction. In *ECAI*, pages 385–391, Stockholm, Sweden, 1990.

[16] Hirofumi Katsuno and Alberto O. Mendelzon. A unified view of propositional knowledge base updates. In N. S. Sridharan, editor, *the 11th International Joint Conference on Artificial Intelligence, IJCAI-89*, pages 1413–1419, Detroit, Michigan, USA, 1989. Morgan Kaufmann.

[17] Hirofumi Katsuno and Alberto O. Mendelzon. On the difference between updating a knowledge base and revising it. In *KR'91*, pages 387–394, Cambridge, Massachusetts, USA, 1991. Morgan Kaufmann Publishers.

[18] Hirofumi Katsuno and Alberto O. Mendelzon. Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 52(3):263–294, 1991.

[19] Vladimir Lifschitz, David Pearce, and Agustín Valverde. Strongly equivalent logic programs. *ACM Transactions on Computational Logic*, 2(4):526–541, October 2001.

[20] Mauricio Osorio and Victor Cuevas. Updates in answer set programming: An approach based on basic structural properties. *Journal of Theory and Practice of Logic Programming*, 7(4):451–479, 2007.

[21] Mauricio Osorio, Juan Antonio Navarro, and José Arrazola. Equivalence in answer set programming. In Alberto Pettorossi, editor, *Logic Based Program Synthesis and Transformation:* 11^{th} *International Workshop; selected papers / LOPSTR 2001, Paphos, Cyprus*, LNCS 2372, pages 57–75, Paphos, Cyprus, November 2001. Springer-Verlag Berlin.

[22] Chiaki Sakama and Katsumi Inoue. An abductive framework for computing knowledge base updates. *Theory and Practice of Logic Programming*, 3(6):671–715, 2003.

[23] Fernando Zacarías, Mauricio Osorio, Juan C. Acosta-G., and Jürgen Dix. Updates in Answer Set Programming Based on Structural Properties. In Sheila McIlraith, Pavlos Peppas, and Michael Thielscher, editors, *7th International Symposium on Logical Formalizations of Commonsense Reasoning*, pages 213–219, Corfu, Greece, May 2005. Fakultät Informatik, TU-Dresden, ISSN 1430-211X.