AGM Postulates in Answer Sets

J. C. Acosta Guadarrama*

Institute of Computer Science,
TU-Clausthal, Germany
jguadarrama@gmail.com

Abstract. Revising and updating beliefs and knowledge bases is an important topic in knowledge representation and reasoning that requires a solid theoretical basis. As a result, various researchers have proposed Answer Set Programming as one of their key components to set up their approaches. In the need to satisfy more general principles, this paper presents a new characterisation of a semantics. It consists in performing updates of *epistemic states* that meets well-accepted *AGM revision postulates*. Besides the formalism of properties that this framework shares with other equivalent update semantics, this proposal is also supported by a solver prototype as an important component of logic programming and automatic testbed of its declarative version. The solver may help compute agent's knowledge bases for more complex potentially-industrial applications and frameworks.

1 Abductive Programs and MGAS

Abduction is an alternative process to *deductive reasoning* in Classical Logic [4], whose formal definitions are omitted due to page-limit reasons. This simple and strong framework is the main core of a solid foundation for the update formulation, presented in the following sections.

2 Updating Epistemic States

One of the latest proposals to meet most *well-accepted principles* for updates at *the object level* and in Minimal Generalised Answer Sets (MGAS) was first introduced in [3]. Such a proposal introduces a flexible foundation to set up the needed models for the desired properties.

A deep analysis of the problem, the solution, *justification, basic model-oriented properties* and *comparison with other approaches* are available in [3]. By now, let us briefly introduce the semantics by a *characterisation in Belief Revision*.

The semantics is formally expressed with the following set of definitions, borrowed and slightly modified from [3] to make it simpler and precise.

Formally, an *α-relaxed rule* is a rule ρ that is *weakened* by a default-negated atom α in its body: $\mathsf{Head}(\rho) \leftarrow \mathsf{Body}(\rho) \cup \{\text{not } \alpha\}$. In addition, an *$\alpha$-relaxed program* is a set of α-relaxed rules.

* This project is supported by a CONACYT Doctorate Grant.

A *generalised program* is a set of rules of form $\ell \leftarrow \top \mid \ell \in \mathcal{A}^*$, where $\mathcal{A}^*$ is a given set of literals.

As a consequence, *updating a program with another* consists in transforming an ordered pair of programs into a *single abductive program*, as follows.

Definition 1 ($\circ_o$-update Program). *Given an updating pair of extended logic programs, denoted as $\Pi_1 \circ_o \Pi_2$, over a set of atoms $\mathcal{A}$; and a set of unique abducibles $\mathcal{A}^*$, such that $\mathcal{A} \cap \mathcal{A}^* = \emptyset$; and the abductive program $\Pi_{\mathcal{A}^*} = \langle \Pi' \cup \Pi_2, \mathcal{A}^* \rangle$ with its corresponding α-relaxed program Π' such that $\alpha \in \mathcal{A}^*$, and its corresponding MGAS's. Its* $\circ_o$-update program *is $\Pi' \cup \Pi_2 \cup \Pi_G$, where Π_G is a generalised program of $\mathsf{M} \cap \mathcal{A}^*$ for some MGAS M of $\Pi_{\mathcal{A}^*}$ and $\circ_o$ is the corresponding update operator.*

Last, the associated *models $\mathcal{S}$ of an epistemic state* (an updating pair) corresponds to the answer sets of a $\circ_o$-update program as follows.

Definition 2 ($\circ_o$-update Answer Set). *Let $\mathbf{\Pi}_{\circ_o} = (\Pi_1 \circ_o \Pi_2)$ be an update pair over a set of atoms $\mathcal{A}$. Then, $\mathcal{S} \subseteq \mathcal{A}$ is an* $\circ_o$-update answer set of $\mathbf{\Pi}_{\circ_o}$ *if and only if $\mathcal{S} = \mathcal{S}' \cap \mathcal{A}$ for some answer set $\mathcal{S}'$ of its $\circ_o$-update program.*

3 $\circ_o$-Properties

The properties of this simple formulation are the main result of this current semantics for *successive updates* of *epistemic states.*

3.1 $\circ_o$-Principles

One of the contributions of this paper is a particular *interpretation and characterisation of AGM [1]reformulation* (R∘1)–(R∘6), due to [2], as a main foundation to *revise* logic programs in ASP.

My own interpretation of postulates (R ∘ 1)–(R ∘ 6) corresponds to postulates (RG ∘ 1)–(RG ∘ 6) below:

(**RG ∘ 1**) $\Pi_1 \subseteq \Pi \circ \Pi_1$.
(**RG ∘ 2**) If $\Pi \cup \Pi_1$ is consistent, then $\Pi \circ \Pi_1 \equiv \Pi \cup \Pi_1$.
(**RG ∘ 3**) If Π_1 is consistent, then $\Pi \circ \Pi_1$ is also consistent.
(**RG ∘ 4**) If $\Pi_1 \equiv_{\mathcal{N}_2} \Pi_2$ then $\Pi \circ \Pi_1 \equiv \Pi \circ \Pi_2$.
(**RG ∘ 5**) $\Pi \circ (\Pi_1 \cup \Pi_2) \subseteq (\Pi \circ \Pi_1) \cup \Pi_2$.
(**RG ∘ 6**) If $(\Pi \circ \Pi_1) \cup \Pi_2$ is consistent, then $(\Pi \circ \Pi_1) \cup \Pi_2 \subseteq \Pi \circ (\Pi_1 \cup \Pi_2)$.

where being *consistent* means having answer sets; "∘" is a generic *revision operator*; and "$\Pi_1 \equiv \Pi_2$" means that Π_1 and Π_2 have the same answer sets. Last, "$\Pi_1 \equiv_{\mathcal{N}_2} \Pi_2$" means that the corresponding translated programs Π_1 and Π_2 into $\mathcal{N}_2$ Nelson's logic theories are equivalent [5].

As an interesting result, let us consider (RG ∘ 3) in order to formulate the following lemma, from which one can get a more general property.

Lemma 1 (weak consistency view). *Suppose Π_0 and Π_1 are ELP's and an updating pair $\Pi_0 \circ_o \Pi_1$ with its corresponding abductive program $\Pi_{\mathcal{A}^*} = \langle \Pi' \cup \Pi_1, \mathcal{A}^* \rangle$. If Π_1 is consistent then $\Pi_{\mathcal{A}^*}$ is consistent.*

Corollary 1 (consistency recovery). *Suppose Π_0 and Π_1 are ELP's. The update $\Pi_0 \circ_o \Pi_1$ is consistent if Π_1 is consistent.*

Corollary 1 also proves to be useful *satisfying belief revision postulates*.

Theorem 1 (RG $\circ$ -properties). *Suppose that Π, Π_1 and Π_2 are ELP. Update operator "$\circ_o$" satisfies properties (RG $\circ$ 1)–(RG $\circ$ 4) and (RG $\circ$ 6).*

Nevertheless, postulate (RG $\circ$ 5) does not hold. As a counterexample, consider the following programs: $\Pi = \{a \leftarrow \top; \neg b \leftarrow \top; \neg c \leftarrow \top\}; \Pi_1 = \{b \leftarrow \top\}; \Pi_2 = \{c \leftarrow \top\}$. This counterexample inverts the direction of the relation.

4 Conclusions

This paper presents work in progress of a *generalisation* of $\circ_o$-operator that satisfies five of the six most suitable *belief-revision postulates* for *updating epistemic states*. As a result, this framework provides a strong theoretical foundation on well-known principles and other fundamental properties.

Finally, as a classical component of Logic Programming, this operator has an implemented *solver prototype* at `http://www2.in.tu-clausthal.de/~guadarrama/updates/o.html`, as an automatic testbed that makes the semantics more *accessible* (in a classroom, i.e.), and potential component for further more complex prototypes in administration of (toy?) knowledge systems, with precise properties. Further details, examples and proofs are coming up in an extended version.

Bibliography

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

[2] Darwiche, A. and Pearl, J. 1994. On the logic of iterated belief revision. In *Proceedings of the fifth Conference on Theoretical Aspects of Reasoning about Knowledge*, R. Fagin, Ed. Morgan Kaufmann, Pacific Grove, CA, 5–23.

[3] Guadarrama, J. C. 2007. Maintaining knowledge bases at the object level. In *Special Session of the 6th International MICAI Conference*, P. Kellenberger, Ed. IEEE Computer Society (to appear), Aguascalientes, Mexico. ISBN: 0-7695-2722-1.

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

[5] Lifschitz, V., Pearce, D., and Valverde, A. 2001. Strongly equivalent logic programs. *ACM Transactions on Computational Logic 2,* 4 (October), 526–541.