TU Clausthal
Clausthal University of Technology

J. C. Acosta Guadarrama
Institute of Computer Science, TU-Clausthal
Julius-Albert-Str. 4, D-38678 Clausthal-Zellerfeld, Germany
jguadarrama@gmail.com

On Updates of Epistemic States

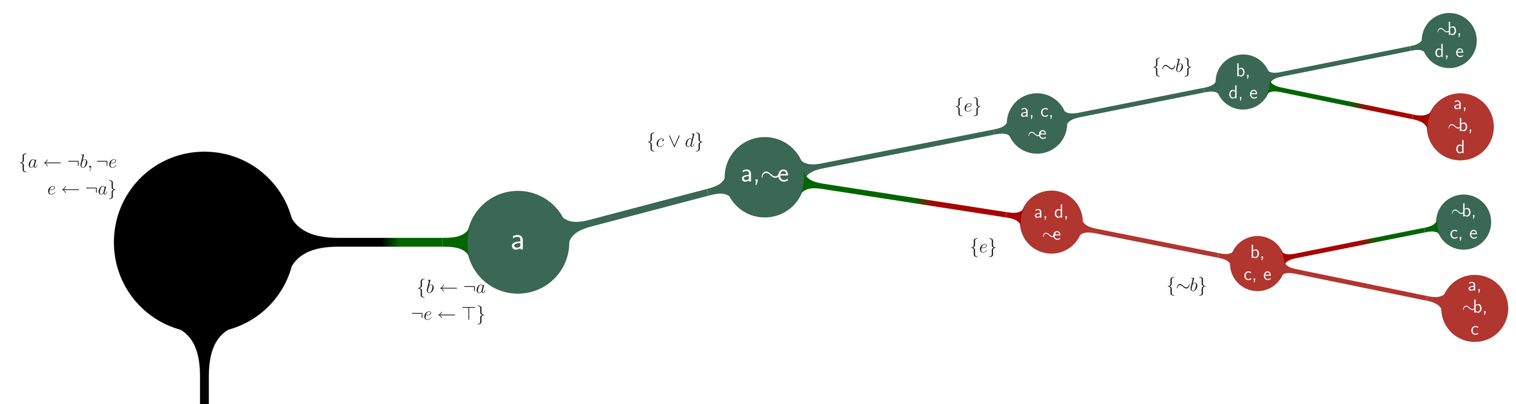
Belief Change under Incomplete Information

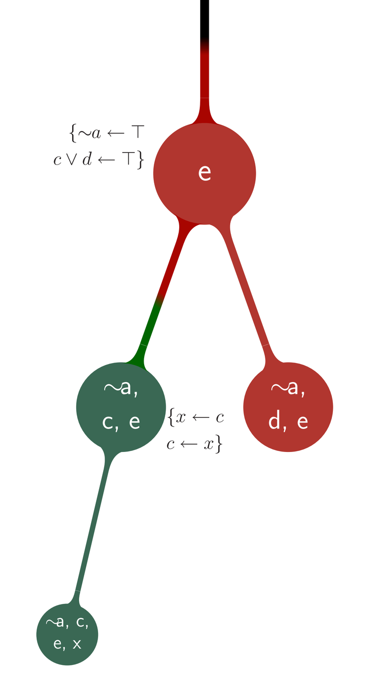
Abstract

In this dissertation I present some aspects of belief-change theory and representation of knowledge as one of the main theoretical basis to formulate semantics for updates of logic programs. Firstly, there is an introduction to relevant principles and postulates, like the classical belief-revision formulation and a following proposal to make a difference between belief revision and updates. Next, there is a survey of some few proposals to update logic programs that are the main motivation for this thesis. Finally, I present a progressive approach that overrides the problems pointed out in other alternatives and that meets most of the principles here introduced.

In particular, revising and updating knowledge bases is an important problem in knowledge representation and reasoning that has led to various proposals for updating logic programs, specifically with respect to the well-known answer-sets semantics. However, most of these approaches have been based on the causal rejection principle that leads to counter-intuitive behaviour. The proposed approach in this thesis is a semantics for abduction known as generalised answer sets, which allows to choose potential models without changing the semantics of the original given update programs. With generalised answer sets one can actually formulate semantics for updates that consist in choosing between generalised models that satisfy an intended set of properties and overcome certain problems from other approaches. Two of the main properties the update semantics should manifest are Weak Irrelevance of Syntax and Strong Consistency, which are a keystone to overcome the mentioned problems.

Finally, as an important component of logic programming and as a useful tool in the classroom, this work also provides the research community with online solver prototypes that help close the gap between theory and practice. These labs of automatic testbeds make the semantics more accessible and opens up a path with a solid component for further more-complex prototypes in systems of knowledge management.





1. Problems

Mainly for a syntax-dependency and lack of theoretical foundation.

Example 1. *Suppose an agent who believes that when it is day it is not night and vice versa, and that there are stars when it is night and when there are no clouds. Finally, that at the current moment it is a fact that there are no stars. This simple story may be coded into* Π_1 *as follows:*

$$\Pi_1 = \{ \begin{array}{l} day \leftarrow \neg night \\ night \leftarrow \neg day \\ stars \leftarrow night, \neg cloudy \\ \sim stars \end{array} \}$$

whose unique answer set is $\{day, \sim stars\}$*. Later, the agent acquires new information stating that* stars *and* constls *(constellations) are the same thing, as coded in* Π_2*. As soon as the agent updates* Π_1 *with program*

$$\Pi_2 = \{stars \leftrightarrow constls\}$$

the expanded alphabet *of the two programs contains only one new* extra atom *with respect to* Π_1*:* constls*. As the model of* Π_2 *is obviously the* empty answer set*,* constls *is considered synonym of* stars *by means of* Π_2*, and thus the update should not change the* original beliefs*.*

Eiter's et alia —[2]: the corresponding semantics gives counterintuitive update answer sets, $\{day, \sim stars\}$ and also $\{night, stars, constls\}$.

DyLP—[1]: the resulting dynamic stable models is $\{day\}$ and not $\{night, stars, constls\}$.

Sakama-Inoue's —[11]: This semantics computes $\{day, \sim stars\}$ only, as one would expect.

Zhang's —[12]: The unique answer set in this semantics, $\{day, \sim stars\}$, coincides with common intuition.

Example 2. *Suppose an initial knowledge base* $\Pi_0 = \{(c \leftarrow r), (r \leftarrow \top)\}$ *updated with* $\Pi_1 = \{\text{not}\, r \leftarrow \text{not}\, c\}$*. Firstly, the initial generalised stable* **model** *of* Π_0 *is* $\{c, r\}$*, and one would expect no changes after the update. However, the update* $\mathcal{P} = \Pi_0 \oplus_R \Pi_1$ *has the* **extra model** $\mathcal{M} = \{not\_c, not\_r\}$ *because* $\mathcal{M} = \{not\_c, not\_r\}$*; Rej*$(\mathcal{P}, \mathcal{M}) = \{r \leftarrow \top\}$*; Def*$(\mathcal{P}, \mathcal{M}) = \{not\_c\}$*; and least*$[(\Pi_0 \cup \Pi_1) \setminus Rej(\mathcal{P}, \mathcal{M}) \cup Def(\mathcal{P}, \mathcal{M})] = \{not\_c, not\_r\} = \mathcal{M}$*.*

Eiter's et alia —[2]: The semantics gives the update answer set $\{c, r\}$ and an extra $\{\sim r\}$.

DyLP—[1]: The resulting dynamic stable models are also both $\{c, r\}$ and the extra $\{not\_c, not\_r\}$.

Sakama-Inoue's —[11]: This semantics computes just $\{c, r\}$, as one would expect.

Zhang's —[12]: This semantics computes just $\{c, r\}$.

DyLP = principles + logic + control

2. Proposal

LOGIC

A1-$\mathcal{N}$: $\sim(\mathfrak{A} \wedge \mathfrak{B}) \equiv (\sim\mathfrak{A} \vee \sim\mathfrak{B})$
A2-$\mathcal{N}$: $\sim(\mathfrak{A} \vee \mathfrak{B}) \equiv (\sim\mathfrak{A} \wedge \sim\mathfrak{B})$
A3-$\mathcal{N}$: $\sim(\mathfrak{A} \supset \mathfrak{B}) \equiv (\mathfrak{A} \wedge \sim\mathfrak{B})$
A4-$\mathcal{N}$: $\sim\sim\mathfrak{A} \equiv \mathfrak{A}$
A5-$\mathcal{N}$: $\sim\neg\mathfrak{A} \equiv \mathfrak{A}$
A6-$\mathcal{N}$: $\sim\mathfrak{A} \supset \neg\mathfrak{A}$, for an atomic $\mathfrak{A}$.
Plus: Heytig's Intuitionistic-logic axioms.

PRINCIPLES

(**RG** ∘ **1**) $\Pi_1 \subseteq \Pi \circ \Pi_1$.
(**RG** ∘ **2**) If $\Pi \cup \Pi_1$ is consistent, then $\Pi \circ \Pi_1 \equiv_{\mathrm{ASP}} \Pi \cup \Pi_1$.
(**RG** ∘ **3**) If Π_1 is consistent, then $\Pi \circ \Pi_1$ is also consistent.
(**RG** ∘ **4**) If $\Pi_x = \Pi_y$ and $\Pi_1 \equiv_{\mathcal{N}_2} \Pi_2$ then $\Pi_x \circ \Pi_1 \equiv_{\mathrm{ASP}} \Pi_y \circ \Pi_2$.
(**RG** ∘ **4′**) If $\Pi_1 \equiv_{\mathcal{N}_2} \Pi_2$ then $\Pi \circ \Pi_1 \equiv_{\mathrm{ASP}} \Pi \circ \Pi_2$.
(**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 $(\Pi \circ \Pi_1) \cup \Pi_2 \subseteq \Pi \circ (\Pi_1 \cup \Pi_2)$.

CONTROL

- Answer-set Programming, $\mathcal{L}_{\mathrm{ASP}}$
- An *α-relaxed rule* is a rule ρ that is *weakened* by a default-negated atom α in its body: Head$(\rho) \leftarrow$ Body$(\rho) \cup \{\neg\alpha\}$. In addition, an *α-relaxed program* is a set of α-relaxed rules.
- On the other hand, 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.
- Semantics for updates

Definition 1 ($\otimes_o$**-update Program).** *Given an updating pair of extended logic programs, denoted as* $\Pi_1 \otimes_o \Pi_2$*, 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 α-relaxed program* Π' *from* Π_1*, such that* $\alpha \in \mathcal{A}^*$*; and the abductive program* $\Pi_{\mathcal{A}^*} = \langle \Pi' \cup \Pi_2, \mathcal{A}^* \rangle$*. Its* $\otimes_o$*-update* program *is* $\Pi' \cup \Pi_2 \cup \Pi_G$*, where* Π_G *is a generalised program of* $\mathcal{M} \cap \mathcal{A}^*$ *for some* minimal generalised answer set $\mathcal{M}$ *of* $\Pi_{\mathcal{A}^*}$ *and* $\otimes_o$ *is the corresponding update operator.*

3. Results

$\otimes_o$**-SP-2, Initialisation [2]:** $\emptyset \otimes_o \Pi \equiv \Pi$.
This property states that the update of an initial empty knowledge base yields just the update itself.

$\otimes_o$**-SP-3, Inertia:** If Π is consistent, $\Pi \otimes_o \emptyset \equiv \Pi$.
A consistent theory is in effect unless new *evidence* states otherwise.

$\otimes_o$**-SP-4, Idempotence [2]:** $\Pi \otimes_o \Pi \equiv \Pi$.
This property means that the update of program Π with itself has no effect.

$\otimes_o$**-SP-6, Non-interference, WNI: [2]:** If Π_1 and Π_2 are programs defined over *disjoint alphabets*, and either both of them have answer sets or not, then $\Pi_1 \otimes_o \Pi_2 \equiv \Pi_2 \otimes_o \Pi_1$.
This property is a specialisation from [2] and implies that the *order of updates* that do not interfere with each other, does not matter.

$\otimes_o$**-SP-7, Augmented Update [2]:** If $\Pi_1 \subseteq \Pi_2$ then $\Pi_1 \otimes_o \Pi_2 \equiv \Pi_2$.
Updating with additional rules makes the previous update obsolete.

$\otimes_o$**-SP-8, Strong Consistency, SC:** If $\Pi_1 \cup \Pi_2$ is *consistent*, then $\Pi_1 \otimes_o \Pi_2 \equiv \Pi_1 \cup \Pi_2$.
The update coincides with the union when $\Pi_1 \cup \Pi_2$ is consistent.

$\otimes_o$**-SP-9, Weak Irrelevance of Syntax, WIS:** Let Π, Π_1, and Π_2 be logic programs under the same language. If $Trans_{\mathcal{N}_2}(\Pi_1) \equiv_{\mathcal{N}_2} Trans_{\mathcal{N}_2}(\Pi_2)$ then $\Pi \otimes_o \Pi_1 \equiv \Pi \otimes_o \Pi_2$.
It means that if we update a program Π with Π_1 or with Π_2, the result should depend upon the *logical contents* of Π_1 and Π_2, rather than the particular syntax to spell them.

References

[1] 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.

[2] 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.

[3] Juan C. Guadarrama. Implementing knowledge update sequences. In Alexander Gelbukh and Angel Kuri Morales, editors, *MICAI 2007: Advances in Artificial Intelligence*, volume 4827 of *LNCS*, pages 260–270, Aguascalientes, Mexico, November 2007. Springer-Verlag.

[4] Juan C. Guadarrama. 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.

[5] Juan C. Guadarrama. A system to maintain knowledge bases of intelligent agents in DLV's weak constraints. In A. P. del Pobil, editor, *Artificial Intelligence and Soft Computing*, number 628, Palma de Mallorca, Spain, September 2008. IASTED/ACTA Press. ISBN: 978-0-88986-755-0.

[6] Juan C. Guadarrama. Update operation in ASP revisited. ISSN: 1860-8477 IfI-08-12, Institute für Informatik, TU-Clausthal, Clausthal, Germany, December 2008.

[7] Juan C. Guadarrama. Maintaining sequences of knowledge bases in ASP. ISSN: 1860-8477 IfI-09-10, Institute für Informatik, TU-Clausthal, Germany, —to appear— 2009.

[8] Juan C. Guadarrama, Jürgen Dix, and Mauricio Osorio. Update sequences in Generalised Answer Set Programming based on structural properties. In Patrick Kellenberger, editor, *Special Session of the 5th International MICAI Conference*, pages 32–41, Mexico City, Mexico, 2006. IEEE Computer Society. ISBN: 0-7695-2722-1.

[9] Juan C. Guadarrama, Jürgen Dix, Mauricio Osorio, and Fernando Zacarías. 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, ISSN 1430-211X.

[10] 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.

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

[12] Yan Zhang. Logic program-based updates. *ACM Transactions on Computational Logic*, 7(3):421–472, July 2006.

4. Comparison

Operator:	$\oplus_R$ [1]	$\diamond_{\mathrm{SI}}$ [11]	$\oplus_2$ [10]	$\oplus_3$ [10]	$\diamond_{\mathrm{Z}}$ [12]	$\triangleleft$ [2]	$\oslash$ [6, 9]	$\otimes$ [3, 8]	$\otimes'_o$ [5, 7]	$\otimes'$ [3, 8]	$\otimes_o$ [4, 5]
Augt. Upd.			✓	×		×	✓	✓	✓	✓	✓
Consy. Res.		✓			×		✓	✓	✓	✓	✓
Idempotence			✓	✓		✓	✓	✓	✓	✓	✓
Inertia		✓					✓	✓	✓	✓	✓
Initialitation		×	✓	✓		✓	✓	✓	✓	×	✓
Integrity Constraints		✓	×	×	×	×	×	✓	✓	✓	✓
Object level	×	✓	×	×	✓	×	×	×	✓	×	✓
Conflicting I.	×	×	×	×	✓	×	×	×	×	×	×
$\mathcal{L}_{\mathrm{ELP}}$	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
$\mathcal{L}_{\mathrm{GLP}}$	✓	×	×	×	×	×	×	×	×	×	×
Min.C.		✓			✓		✓	✓	✓	✓	✓
Multiple Updates	✓	✓	×	×	×	✓	×	✓	✓	✓	✓
SC	×		×	×	×	×	✓	✓	✓	✓	✓
WC							✓	✓	✓	✓	✓
WIS	×			×		×	✓	✓	✓	✓	✓
WNI			×	✓		✓	✓	✓	✓	✓	✓
(R ∘ 1)			✓	✓		✓	✓	✓	✓	×	✓
(R ∘ 2)			×	×		×	✓	✓	✓	✓	✓
(R ∘ 3)							✓	✓	✓	✓	✓
(R ∘ 4)			✓	×		×	✓	✓	✓	✓	✓
(R ∘ 5)				✓		✓	×				×
(R ∘ 6)				×		×	✓				✓
(PK ⋄ 1)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(PK ⋄ 3)			✓	✓		✓	✓	✓	✓	✓	✓
(PK ⋄ 5)			✓	✓		✓	✓	✓	✓	×	✓
Assoc.							×				✓
Distrib.							✓				✓
W.Comm.							✓				✓