TLP **14** (6): 869–907, 2014. © Cambridge University Press 2013
doi:10.1017/S1471068413000100 First published online 7 August 2013

The rise and fall of semantic rule updates based on SE-models*

MARTIN SLOTA and JOÃO LEITE
CENTRIA and Departamento de Informática, Universidade Nova de Lisboa, 2829-516 Caparica, Portugal
(*e-mail:* `jleite@fct.unl.pt`)

submitted 4 November 2011; revised 6 December 2012; accepted 6 June 2013

Abstract

Logic programs under the stable model semantics, or answer-set programs, provide an expressive rule-based knowledge representation framework, featuring a formal, declarative and well-understood semantics. However, handling the evolution of rule bases is still a largely open problem. The Alchourrón, Gärdenfors and Makinson (AGM) framework for belief change was shown to give inappropriate results when directly applied to logic programs under a non-monotonic semantics such as the stable models. The approaches to address this issue, developed so far, proposed update semantics based on manipulating the syntactic structure of programs and rules.

More recently, AGM revision has been successfully applied to a significantly more expressive semantic characterisation of logic programs based on SE-models. This is an important step, as it changes the focus from the evolution of a syntactic representation of a rule base to the evolution of its semantic content.

In this paper, we borrow results from the area of belief update to tackle the problem of updating (instead of revising) answer-set programs. We prove a representation theorem which makes it possible to constructively define any operator satisfying a set of postulates derived from Katsuno and Mendelzon's postulates for belief update. We define a specific operator based on this theorem, examine its computational complexity and compare the behaviour of this operator with syntactic rule update semantics from the literature. Perhaps surprisingly, we uncover a serious drawback of all rule update operators based on Katsuno and Mendelzon's approach to update and on SE-models.

KEYWORDS: belief update, answer-set programs, rule update, SE-models, support, literal inertia

1 Introduction

Answer-Set Programming (ASP) (Gelfond and Lifschitz 1988; Baral 2003) is now widely recognised as a valuable approach to knowledge representation and reasoning, mostly due to its simple and well-understood declarative semantics, its rich expressive power, and the existence of efficient implementations.

* This is an extended version of Slota and Leite (2010).

However, the dynamic character of many applications that can benefit from ASP calls for the development of ways to deal with the evolution of answer-set programs and the inconsistencies that may arise.

The problems associated with knowledge evolution have been extensively studied, over the years, by different research communities, namely in the context of Classical Logic, and in the context of Logic Programming.

The former have been inspired, to a large extent, by the seminal work of Alchourrón, Gärdenfors and Makinson (AGM), who proposed a set of desirable properties of belief change operators, now called *AGM postulates* (Alchourrón *et al.* 1985). Subsequently, *update* and *revision* have been distinguished as two very related but ultimately different belief change operations (Keller and Winslett 1985; Winslett 1990; Katsuno and Mendelzon 1991). While revision deals with incorporating new information about a *static* world, update takes place when changes occurring in a *dynamic* world are recorded. Katsuno and Mendelzon (1991) formulated a separate set of postulates for update, now known as *KM postulates.*

Both AGM and KM postulates were later studied in the context of Logic Programming, only to find that their formulations based on a non-monotonic semantics, such as the answer sets, are inappropriate (Eiter *et al.* 2002). Like many belief change operators, earlier methods used to tackle rule updates were based on literal inertia (Alferes and Pereira 1996) but proved not sufficiently expressive. This led to the development of rule update semantics based on different intuitions, principles and constructions, when compared to their classical counterparts. For example, the introduction of the *causal rejection principle* (Leite and Pereira 1998) motivated a line of work on several rule update semantics (Alferes *et al.* 2000; Eiter *et al.* 2002; Leite 2003; Alferes *et al.* 2005; Osorio and Cuevas 2007), all of them with a strong syntactic flavour. Other approaches tackle rule updates by employing syntactic transformations and other methods, such as abduction (Sakama and Inoue 2003), forgetting (Zhang and Foo 2005), prioritisation (Zhang 2006), preferences (Delgrande *et al.* 2007) or dependencies on default assumptions (Šefránek 2006; Šefránek 2011; Krümpelmann and Kern-Isberner 2010).

Though useful in a number of practical scenarios (Alferes *et al.* 2003; Saias and Quaresma 2004; Siska 2006; Ilic *et al.* 2008; Slota *et al.* 2011), it turned out that most of these semantics exhibit undesirable behaviour. For example, except for the semantics proposed in Alferes *et al.* (2005) and Šefránek (2011), a tautological update may influence the result under all of these semantics, a behaviour that is highly undesirable when considering knowledge updates. Other kinds of irrelevant updates are even more problematic and subject of ongoing research (Šefránek 2006, 2011). But, more important, the common feature of all of these semantics is that they make heavy use of the syntactic structure of programs and rules, making any analysis of their semantic properties a daunting task.

Recently, AGM revision was reformulated in the context of Logic Programming in a manner analogous to belief revision in classical propositional logic, and specific revision operators for logic programs were investigated (Osorio and Cuevas 2007; Delgrande *et al.* 2008). Central to this novel approach are *SE-models* (Turner 2003) which provide a monotonic semantic characterisation of logic programs that is strictly more expressive than the answer-set semantics. Furthermore, two programs have the same set of SE-models if and only if they are strongly equivalent (Lifschitz *et al.* 2001), which means that programs $\mathscr{P}, \mathscr{Q}$ with the same set of SE-models can be modularly replaced by one another, even in the presence of additional rules, without affecting the resulting answer sets.

Indeed, these results constitute an important breakthrough in the research of answer-set program evolution. They change the focus from the syntactic representation of a program, where not all rules and literal occurrences are necessarily relevant to the meaning of the program as a whole, to its semantic content, i.e. to the information that the program is intended to represent.

In this paper, we follow a similar path, but to tackle the problem of answer-set program *updates*, instead of *revision* as in Delgrande *et al.* (2008).

Using *SE-models*, we adapt the KM postulates to answer-set program updates and prove a representation theorem that provides a constructive characterisation of rule update operators satisfying the postulates, making it possible to define and evaluate any operator satisfying the postulates using an intuitive construction. We show how this constructive characterisation can be used by defining a concrete answer-set program update operator that can be seen as a counterpart of Winslett's belief update operator (Winslett 1990) which satisfies the KM postulates and is commonly used in the literature.

However, while investigating the operator's properties, we uncover a serious drawback which, as it turns out, extends to all answer-set program update operators based on SE-models and Katsuno and Mendelzon's approach to updates. This finding is very important as it guides the research on updates of answer-set programs away from the purely semantic approach materialised in AGM and KM postulates or, alternatively, to the development of semantic characterisations of answer-set programs, richer than SE-models, that are appropriate for describing their dynamic behaviour.

The remainder of this paper is structured as follows. In Section 2, we introduce the formal concepts that are necessary throughout the rest of the paper. Section 3 contains the reformulation of KM postulates for logic program updates and the representation theorem that establishes a general constructive characterisation of rule update operators obeying the postulates. We also show how this theorem can be used by defining a specific rule update operator that satisfies the postulates and we examine the computational complexity of query answering for this operator. In Section 4, we further analyse the previously defined operator and establish that all semantic rule update operators based on SE-models exhibit an undesired behaviour. We summarise our findings in Section 5.

2 Preliminaries

We consider a propositional language over a finite set of propositional variables $\mathcal{A}$ and the usual set of propositional connectives to form propositional formulae. An *objective literal* is either an atom p or its negation $\neg p$. A *Horn clause* is a disjunction of at most one atom and zero or more negated atoms; a *Horn formula* is a conjunction of Horn clauses.

A (propositional) interpretation is any subset of $\mathcal{A}$ and the set of all interpretations is $\mathfrak{I} = 2^{\mathcal{A}}$. We use the standard semantics for propositional formulae and denote the set of models of a formula ϕ by $[\![\phi]\!]$. We also write $J \models \phi$ if $J \in [\![\phi]\!]$. We say that a formula ϕ is *complete* if $[\![\phi]\!]$ is a singleton set. For formulae ϕ, ψ we say that ϕ *is equivalent to* ψ, denoted by $\phi \equiv \psi$, if $[\![\phi]\!] = [\![\psi]\!]$, and that ϕ *entails* ψ, denoted by $\phi \models \psi$, if $[\![\phi]\!] \subseteq [\![\psi]\!]$. As we are dealing with the finite case, every knowledge base can be expressed by a single formula.

2.1 Belief update

Update is a belief change operation that brings a knowledge base *up to date* when the *world described by it changes* (Keller and Winslett 1985; Katsuno and Mendelzon 1991). Formally, a belief update operator is a function that takes two formulae, representing the original knowledge base and its update, as arguments and returns a formula representing the updated knowledge base. To further specify the desired properties of update operators, the following eight postulates for a belief update operator $\diamond$ and formulae ϕ, ψ, μ, ν were proposed in Katsuno and Mendelzon (1991):

(B1) $\phi \diamond \mu \models \mu$.

(B2) If $\phi \models \mu$, then $\phi \diamond \mu \equiv \phi$.

(B3) If $[\![\phi]\!] \neq \emptyset$ and $[\![\mu]\!] \neq \emptyset$, then $[\![\phi \diamond \mu]\!] \neq \emptyset$.

(B4) If $\phi \equiv \psi$ and $\mu \equiv \nu$, then $\phi \diamond \mu \equiv \psi \diamond \nu$.

(B5) $(\phi \diamond \mu) \wedge \nu \models \phi \diamond (\mu \wedge \nu)$.

(B6) If $\phi \diamond \mu \models \nu$ and $\phi \diamond \nu \models \mu$, then $\phi \diamond \mu \equiv \phi \diamond \nu$.

(B7) If ϕ is complete, then $(\phi \diamond \mu) \wedge (\phi \diamond \nu) \models \phi \diamond (\mu \vee \nu)$.

(B8) $(\phi \vee \psi) \diamond \mu \equiv (\phi \diamond \mu) \vee (\psi \diamond \mu)$.

Katsuno and Mendelzon also proved an important representation theorem that makes it possible to define and evaluate any operator satisfying these postulates using an intuitive construction. It is based on treating the models of a knowledge base as possible real states of the modelled world. An update of an original knowledge base ϕ is performed by modifying each of its models as little as possible to make it consistent with new information in the update μ, obtaining a new set of interpretations – the models of the updated knowledge base. More formally,

$$[\![\phi \diamond \mu]\!] = \bigcup_{I \in [\![\phi]\!]} \mathsf{incorporate}([\![\mu]\!], I),$$

where $\mathsf{incorporate}(\mathcal{M}, I)$ returns the members of $\mathcal{M}$ closer to I. A natural way of defining $\mathsf{incorporate}(\mathcal{M}, I)$ is by assigning an order $\leqslant^I$ over $\mathfrak{I}$ to each interpretation I and taking the minima of $\mathcal{M}$ w.r.t. $\leqslant^I$, i.e. $\mathsf{incorporate}(\mathcal{M}, I) = \min(\mathcal{M}, \leqslant^I)$. In the following we first formally establish the concept of an *order assignment*; thereafter we define when an update operator is *characterised by* such an assignment.

Given a set $\mathscr{S}$, a *preorder over* $\mathscr{S}$ is a reflexive and transitive binary relation over $\mathscr{S}$; a *strict preorder over* $\mathscr{S}$ is an irreflexive and transitive binary relation over $\mathscr{S}$; a *partial order over* $\mathscr{S}$ is a preorder over $\mathscr{S}$ that is antisymmetric. Given a preorder $\leqslant$ over $\mathscr{S}$, we denote by $<$ the strict preorder induced by $\leqslant$, i.e. $s < t$ if and only if $s \leqslant t$ and not $t \leqslant s$. For any subset $\mathscr{T}$ of $\mathscr{S}$, the set of *minimal elements of* $\mathscr{T}$ *w.r.t.* $\leqslant$ is

$$\min(\mathscr{T}, \leqslant) = \{\, s \in \mathscr{T} \mid \neg\exists t \in \mathscr{T} : t < s \,\}.$$

Definition 1 (*Order assignment*)
Let $\mathscr{S}$ be a set. A *preorder assignment over* $\mathscr{S}$ is any function ω that assigns a preorder $\leqslant^s_\omega$ over $\mathscr{S}$ to each $s \in \mathscr{S}$. A *partial order assignment over* $\mathscr{S}$ is any preorder assignment ω over $\mathscr{S}$ such that $\leqslant^s_\omega$ is a partial order over $\mathscr{S}$ for every $s \in \mathscr{S}$.

Definition 2 (*Belief update operator characterised by an order assignment*)
Let $\diamond$ be a belief update operator and ω a preorder assignment over $\mathfrak{I}$. We say that $\diamond$ is *characterised by* ω if for all formulae ϕ, μ,

$$[\![\phi \diamond \mu]\!] = \bigcup_{I \in [\![\phi]\!]} \min\left([\![\mu]\!], \leqslant^I_\omega\right). \tag{1}$$

A natural condition to impose on the assigned orders is that every interpretation be the closest to itself. This is captured by the notion of a *faithful* order assignment:

Definition 3 (*Faithful order assignment; Katsuno and Mendelzon 1991*)
A preorder assignment ω over $\mathfrak{I}$ is *faithful* if for every interpretation I the following condition is satisfied:

For every $J \in \mathfrak{I}$ with $J \neq I$ it holds that $I <^I_\omega J$.

The representation theorem of Katsuno and Mendelzon (1991) states that operators characterised by faithful order assignments are exactly those that satisfy the KM postulates.

Theorem 4 (*Representation theorem for belief updates; Katsuno and Mendelzon 1991*)

Let $\diamond$ be a belief update operator. Then the following conditions are equivalent:

(a) The operator $\diamond$ satisfies conditions (B1)–(B8).
(b) The operator $\diamond$ is characterised by a faithful preorder assignment.
(c) The operator $\diamond$ is characterised by a faithful partial order assignment.

Katsuno and Mendelzon's results provide a framework for belief update operators, each specified on the semantic level by a faithful partial order assignment over $\mathfrak{I}$. The most influential instance of this framework is the *Possible Models Approach* (Keller and Winslett 1985; Winslett 1990), also referred to as *Winslett's belief update semantics*, based on minimising the set of atoms whose truth value changes when an interpretation is updated. Formally, Winslett's partial order assignment W is defined for all interpretations I, J, K by

$$J \leqslant^I_{\mathsf{W}} K \qquad \text{if and only if} \qquad (J \div I) \subseteq (K \div I),$$

where $\div$ denotes set-theoretic symmetric difference. It is not difficult to verify that W is a faithful partial order assignment, so it follows from Theorem 4 that any belief update operator $\diamond$ characterised by W satisfies postulates (B1)–(B8). Note that there is a whole class of operators characterised by W that differ in the syntactic representation of updated belief bases. Insofar as we are interested in the semantic properties of Winslett's updates, it follows from (B4) that it does not matter which operator from this class we pick. This is illustrated in the following example:

Example 5 (*Winslett's belief update semantics*)
Consider the knowledge base $\phi = (p \wedge (q \equiv r))$ and the update $\mu = (q \vee r)$ over the set of atoms $\mathcal{A} = \{ p, q, r \}$. Their sets of models are as follows:

$$\llbracket \phi \rrbracket = \{ \{ p \}, \{ p, q, r \} \},$$
$$\llbracket \mu \rrbracket = \{ \{ q \}, \{ r \}, \{ q, r \}, \{ p, q \}, \{ p, r \}, \{ p, q, r \} \}.$$

When performing an update of ϕ by μ under Winslett's update semantics, equation (1) applies as follows:

$$\llbracket \phi \diamond \mu \rrbracket = \bigcup_{I \in \llbracket \phi \rrbracket} \min \left(\llbracket \mu \rrbracket, \leqslant^I_{\mathsf{W}} \right) = \min \left(\llbracket \mu \rrbracket, \leqslant^{\{ p \}}_{\mathsf{W}} \right) \cup \min \left(\llbracket \mu \rrbracket, \leqslant^{\{ p,q,r \}}_{\mathsf{W}} \right).$$

The models of μ that 'differ least' from $\{ p \}$, in the sense of the order assignment W, are $\{ p, q \}$ and $\{ p, r \}$. Furthermore, since W is faithful, the unique model of μ that is minimally distant from $\{ p, q, r \}$ is $\{ p, q, r \}$ itself. Consequently,

$$\llbracket \phi \diamond \mu \rrbracket = \{ \{ p, q \}, \{ p, r \}, \{ p, q, r \} \}.$$

Note that from the syntactic viewpoint, $\phi \diamond \mu$ can be any formula with the above set of models. Thus, it may for example be the case that $\phi \diamond \mu = (p \wedge (q \vee r))$ while for another operator $\diamond'$, also characterised by W, $\phi \diamond' \mu = ((p \wedge q) \vee (p \wedge r))$.

2.2 Computational complexity of Winslett's update semantics

Computationally, query answering for Winlett's operator, i.e. the problem of deciding whether $\phi \diamond \mu \models \psi$, where $\diamond$ is characterised by W, belongs to the second level of the polynomial hierarchy (Eiter and Gottlob 1992). We formulate this result formally as it later facilitates the study of computational complexity of a newly introduced rule update operator.

Assuming that the reader is familiar with the classes NP and co-NP, we briefly introduce the *polynomial hierarchy* (Meyer and Stockmeyer 1972; Stockmeyer 1976). Its definition relies on the notion of an *oracle*: An oracle for a class of decision problems C can decide any problem in C in just one step of computation. We denote by NP^C the class of decision problems solvable in polynomial time by a non-deterministic Turing machine that can make calls to an oracle for C. The classes Σ^{P}_i

and Π_i^{P} of the polynomial hierarchy are defined inductively as follows: $\Sigma_0^{\mathrm{P}} = \Pi_0^{\mathrm{P}} = \mathrm{P}$ and for all $i \geqslant 0$,

$$\Sigma_{i+1}^{\mathrm{P}} = \mathrm{NP}^{\Sigma_i^{\mathrm{P}}} \qquad \text{and} \qquad \Pi_{i+1}^{\mathrm{P}} = \text{co-}\Sigma_{i+1}^{\mathrm{P}}.$$

In the general case, query answering for Winslett's updates is Π_2^{P}-complete.

Theorem 6 (*Part of Theorem 6.4 in Eiter and Gottlob 1992*)
Let $\diamond$ be a belief update operator characterised by W. Deciding whether $\phi \diamond \mu \models \psi$ for formulae ϕ, μ, ψ is Π_2^{P}-complete. Hardness holds even if ϕ is a conjunction of atoms and ψ is one of the atoms in that conjunction.

However, when dealing only with Horn formulae, the problem drops to the first level of the polynomial hierarchy:

Theorem 7 (*Part of Theorem 7.2 in Eiter and Gottlob 1992*)
Let $\diamond$ be a belief update operator characterised by W. Deciding whether $\phi \diamond \mu \models \psi$ for Horn formulae ϕ, μ, ψ is co-NP-complete. Hardness holds even if ϕ is a conjunction of objective literals and ψ is one of the literals in that conjunction.

2.3 Logic programming

We define the syntax and semantics of logic programs, borrowing some of the notation used in Delgrande *et al.* (2008).

An *atom* is any $p \in \mathcal{A}$. A *literal* is an atom p or its default negation $\sim p$. Given a set of literals S, we introduce the following notation:

$$S^+ = \{\, p \in \mathcal{A} \mid p \in S \,\}, \qquad S^- = \{\, p \in \mathcal{A} \mid \sim p \in S \,\},$$
$$\sim S = \{\, \sim p \mid p \in S \cap \mathcal{A} \,\}, \qquad \neg S = \{\, \neg p \mid p \in S \cap \mathcal{A} \,\}.$$

A *rule* is a pair of sets of literals $\pi = (H(\pi), B(\pi))$. We say that $H(\pi)$ is the *head of* π and $B(\pi)$ is the *body of* π. Usually, for convenience, we write π as

$$H(\pi)^+ ; \sim H(\pi)^- \leftarrow B(\pi)^+, \sim B(\pi)^-. \tag{2}$$

Operators ';' and ',' express disjunctive and conjunctive connectives, respectively. A rule is called a *fact* if its head contains exactly one literal and its body is empty. A fact is *positive* if the literal in its head is an atom. A rule is *non-disjunctive* if its head contains at most one literal; *definite* if it is non-disjunctive and its head and body contain only atoms. A *program* is a set of rules. A program is *non-disjunctive* if all rules inside it are non-disjunctive; *definite* if all rules inside it are definite.

Turning to the semantics, we need to define *answer sets* and *SE-models* of a logic program. We start by defining the more basic notion of a *(classical) model* of a logic program. For every rule π of the form (2) we denote by $\kappa(\pi)$ the propositional formula

$$\bigwedge (B(\pi)^+ \cup \neg B(\pi)^-) \supset \bigvee (H(\pi)^+ \cup \neg H(\pi)^-).$$

For a program $\mathscr{P}$, $\kappa(\mathscr{P}) = \bigwedge_{\pi \in \mathscr{P}} \kappa(\pi)$. An interpretation J is a *model* of a program $\mathscr{P}$, denoted by $J \models \mathscr{P}$, if $J \models \kappa(\mathscr{P})$. We say that $\mathscr{P}$ is *consistent* if it has some classical model.

An interpretation J is an *answer set* of a program $\mathscr{P}$ if it is a subset-minimal model of the *reduct of $\mathscr{P}$ relative to J*:

$$\mathscr{P}^J = \{ H(\pi)^+ \leftarrow B(\pi)^+. \mid \pi \in \mathscr{P} \land H(\pi)^- \subseteq J \land B(\pi)^- \cap J = \emptyset \}.$$

SE*-models* (Turner 2003), based on the non-classical logic of Here-and-There (Heyting 1930; Łukasiewicz 1941; Pearce 1997), provide a monotonic characterisation of logic programs that is expressive enough to capture both their classical models and answer sets. We use SE-models in the following sections to reformulate the KM postulates for belief update in the context of rule updates.

Intuitively, each SE-interpretation assigns *one of three truth values* to every atom. Technically it consists of a pair of propositional interpretations, the first containing atoms that are true and the second containing atoms that are not false. Formally:

Definition 8 (SE*-interpretation; Turner 2003*)
An SE*-interpretation* is a pair of interpretations (I, J) such that $I \subseteq J$. The set of all SE-interpretations is denoted by $\mathfrak{X}$.

SE-models themselves are defined by referring to the *program reduct* used to define answer sets above.

Definition 9 (SE*-model; Turner 2003*)
Let $\mathscr{P}$ be a program. An SE-interpretation (I, J) is an SE*-model of $\mathscr{P}$* if $J \models \mathscr{P}$ and $I \models \mathscr{P}^J$. The set of all SE-models of $\mathscr{P}$ is denoted by $[\![\mathscr{P}]\!]_{\mathsf{SE}}$ and we write $(I, J) \models \mathscr{P}$ if $(I, J) \in [\![\mathscr{P}]\!]_{\mathsf{SE}}$.

Note that $J \models \mathscr{P}$ if and only if $(J, J) \in [\![\mathscr{P}]\!]_{\mathsf{SE}}$, so SE-models capture the classical models of a program. And just like classical models, the set of SE-models of a program is *monotonic*, i.e. larger programs have smaller sets of SE-models. This is one of the important differences between SE-models and the non-monotonic answer sets.

Nevertheless, a program's answer sets, just like its classical models, can be extracted from its set of SE-models. An interpretation J is an answer set of $\mathscr{P}$ if and only if $(J, J) \in [\![\mathscr{P}]\!]_{\mathsf{SE}}$ and no $(I, J) \in [\![\mathscr{P}]\!]_{\mathsf{SE}}$ with $I \subsetneq J$ exists. This implies that programs with the same set of SE-models also have the same answer sets. Moreover, when such programs are augmented with the same set of rules, the resulting programs *still* have the same answer sets. In many situations such a property is desirable as it allows one program to be modularly replaced by another one, even in the presence of additional rules, without affecting the resulting answer sets. It is typically referred to as *strong equivalence* (Lifschitz *et al.* 2001) and the relationship between SE-models and strong equivalence is formally captured as follows:

Proposition 10 (SE*-models and strong equivalence; Turner 2003*)
Let $\mathscr{P}$, $\mathscr{Q}$ be programs. It holds that $[\![\mathscr{P}]\!]_{\mathsf{SE}} = [\![\mathscr{Q}]\!]_{\mathsf{SE}}$ if and only if for every program $\mathscr{R}$, the answer sets of $\mathscr{P} \cup \mathscr{R}$ and $\mathscr{Q} \cup \mathscr{R}$ are the same.

In other words, SE-models exactly capture the concept of strong equivalence. This also explains the origin of the name SE*-models* – 'SE' stands for *strong equivalence*. Based on this result, we define strong equivalence and entailment as follows:

Definition 11 (*Strong equivalence and strong entailment*)
Let $\mathscr{P}$, $\mathscr{Q}$ be programs. We say that $\mathscr{P}$ *is strongly equivalent to* $\mathscr{Q}$, denoted by $\mathscr{P} \equiv_{\mathsf{SE}} \mathscr{Q}$, if $[\![\mathscr{P}]\!]_{\mathsf{SE}} = [\![\mathscr{Q}]\!]_{\mathsf{SE}}$, and that $\mathscr{P}$ *strongly entails* $\mathscr{Q}$, denoted by $\mathscr{P} \models_{\mathsf{SE}} \mathscr{Q}$, if $[\![\mathscr{P}]\!]_{\mathsf{SE}} \subseteq [\![\mathscr{Q}]\!]_{\mathsf{SE}}$.

An important distinguishing property of SE-models that we will need to carefully consider in the following sections is that whenever a program $\mathscr{P}$ has the SE-model (I, J), it also has the SE-model (J, J). More generally, any set of SE-interpretations with this property is referred to as *well-defined* (Delgrande *et al.* 2008).

Definition 12 (*Well-defined set of SE-interpretations; Delgrande et al. 2008*)
For every SE-interpretation $X = (I, J)$ we denote by X^* the SE-interpretation (J, J). A set of SE-interpretations $\mathscr{M}$ is *well-defined* if for every SE-interpretation X, $X \in \mathscr{M}$ implies $X^* \in \mathscr{M}$.

In fact, as pinpointed in the following result, not only is the set of SE-models of a program well-defined, but every well-defined set of SE-interpretations is also the set of SE-models of some program.

Proposition 13 (*Delgrande et al. 2008*)
A set of SE-interpretations $\mathscr{M}$ is well-defined if and only if $\mathscr{M} = [\![\mathscr{P}]\!]_{\mathsf{SE}}$ for some program $\mathscr{P}$.

As a consequence, whenever $I \subsetneq J$, there is no program that has the single SE-model $X = (I, J)$, though there is a program that has the pair of SE-models X, X^*. The following notion of a *basic program* is thus analogous to the concept of a *complete formula* that is used in the formulation of belief update postulate (B7).

Definition 14 (*Basic program*)
We say that a program $\mathscr{P}$ is *basic* if $[\![\mathscr{P}]\!]_{\mathsf{SE}} = \{ X, X^* \}$ for some SE-interpretation X.

Note that a program is basic if either it has a unique SE-model (J, J), or a pair of SE-models (I, J) and (J, J). In the former case, the program exactly determines the truth values of all atoms – the atoms in J are true and the remaining atoms are false. In the latter case, the program makes atoms in I true, the atoms in $J \setminus I$ may either be undefined or true, as long as they all have the same truth value, and the remaining atoms are false.

3 Semantic rule updates based on SE-Models

With the necessary concepts defined, we are ready to step forward and tailor the belief update postulates and operators to the context of logic programs viewed through their sets of SE-models. Since SE-models provide a *monotonic* characterisation of logic programs, the analysis provided in Eiter *et al.* (2002), which showed KM postulates not appropriate for use with non-monotonic semantics, no longer applies. In the following, we reformulate the belief update postulates as well as a constructive characterisation of semantic rule update operators, and finally show a counterpart of the representation theorem for belief updates. The studied operators are semantic in their very nature and in line with KM postulates, in contrast with the traditional syntax-based approaches to rule updates (Leite and Pereira 1998; Alferes *et al.* 2000, 2005; Eiter *et al.* 2002; Sakama and Inoue 2003; Zhang and Foo 2005; Zhang 2006; Delgrande *et al.* 2007; Šefránek 2011).

Similarly, as in the case of belief updates, we liberally define a rule update operator as any function that takes two inputs, the original program and its update, and returns the updated program.

Definition 15 (*Rule update operator*)
A *rule update operator* is a binary function on the set of all programs.

In order to reformulate postulates (B1)–(B8) for logic programs under the SE-model semantics, we first need to specify what a conjunction and disjunction of logic programs is. To this end, we introduce program conjunction and disjunction operators. These are required to assign, to each pair of programs, a program whose set of SE-models is the intersection and union, respectively, of the sets of SE-models of argument programs.

Definition 16 (*Program conjunction and disjunction*)
A binary operator $\dot\wedge$ on the set of all programs is a *program conjunction operator* if for all programs $\mathscr{P}$, $\mathscr{Q}$,

$$[\![\mathscr{P} \dot\wedge \mathscr{Q}]\!]_{\mathsf{SE}} = [\![\mathscr{P}]\!]_{\mathsf{SE}} \cap [\![\mathscr{Q}]\!]_{\mathsf{SE}}.$$

A binary operator $\dot\vee$ on the set of all programs is a *program disjunction operator* if for all programs $\mathscr{P}$, $\mathscr{Q}$,

$$[\![\mathscr{P} \dot\vee \mathscr{Q}]\!]_{\mathsf{SE}} = [\![\mathscr{P}]\!]_{\mathsf{SE}} \cup [\![\mathscr{Q}]\!]_{\mathsf{SE}}.$$

In the following, we assume that some program conjunction and disjunction operators $\dot\wedge$, $\dot\vee$ are given. Note that the program conjunction operator may simply return the union of argument programs; it is the same as the *expansion operator* defined in Delgrande *et al.* (2008). A program disjunction operator can be defined by translating the argument programs into the logic of Here-and-There (Heyting 1930; Łukasiewicz 1941; Pearce 1997), taking their disjunction and transforming the resulting formula back into a logic program (using results from Cabalar and Ferraris 2007).

The final obstacle before we can proceed with introducing the new postulates is the following. We need to substitute the notion of a *complete formula* used in (B7) with a suitable class of logic programs. It turns out that the notion of a *basic program*, as introduced in Definition 14, is a natural candidate for this purpose. While a complete formula is defined as having a unique model, a program is basic if it has either a unique SE-model (J, J), or a pair of SE-models (I, J) and (J, J). The latter case needs to be allowed in order to make the new postulate applicable to SE-interpretations (I, J) with $I \subsetneq J$ because no program has the single SE-model (I, J) (cf. Proposition 13).

The following are the reformulated postulates for a rule update operator $\oplus$ and programs $\mathscr{P}$, $\mathscr{Q}$, $\mathscr{U}$, $\mathscr{V}$:

$(\mathrm{P1})_{\mathsf{SE}}$ $\mathscr{P} \oplus \mathscr{U} \models_{\mathsf{SE}} \mathscr{U}$.

$(\mathrm{P2})_{\mathsf{SE}}$ If $\mathscr{P} \models_{\mathsf{SE}} \mathscr{U}$, then $\mathscr{P} \oplus \mathscr{U} \equiv_{\mathsf{SE}} \mathscr{P}$.

$(\mathrm{P3})_{\mathsf{SE}}$ If $[\![\mathscr{P}]\!]_{\mathsf{SE}} \neq \emptyset$ and $[\![\mathscr{U}]\!]_{\mathsf{SE}} \neq \emptyset$, then $[\![\mathscr{P} \oplus \mathscr{U}]\!]_{\mathsf{SE}} \neq \emptyset$.

$(\mathrm{P4})_{\mathsf{SE}}$ If $\mathscr{P} \equiv_{\mathsf{SE}} \mathscr{Q}$ and $\mathscr{U} \equiv_{\mathsf{SE}} \mathscr{V}$, then $\mathscr{P} \oplus \mathscr{U} \equiv_{\mathsf{SE}} \mathscr{Q} \oplus \mathscr{V}$.

$(\mathrm{P5})_{\mathsf{SE}}$ $(\mathscr{P} \oplus \mathscr{U}) \dot{\wedge} \mathscr{V} \models_{\mathsf{SE}} \mathscr{P} \oplus (\mathscr{U} \dot{\wedge} \mathscr{V})$.

$(\mathrm{P6})_{\mathsf{SE}}$ If $\mathscr{P} \oplus \mathscr{U} \models_{\mathsf{SE}} \mathscr{V}$ and $\mathscr{P} \oplus \mathscr{V} \models_{\mathsf{SE}} \mathscr{U}$, then $\mathscr{P} \oplus \mathscr{U} \equiv_{\mathsf{SE}} \mathscr{P} \oplus \mathscr{V}$.

$(\mathrm{P7})_{\mathsf{SE}}$ If $\mathscr{P}$ is basic, then $(\mathscr{P} \oplus \mathscr{U}) \dot{\wedge} (\mathscr{P} \oplus \mathscr{V}) \models_{\mathsf{SE}} \mathscr{P} \oplus (\mathscr{U} \dot{\vee} \mathscr{V})$.

$(\mathrm{P8})_{\mathsf{SE}}$ $(\mathscr{P} \dot{\vee} \mathscr{Q}) \oplus \mathscr{U} \equiv_{\mathsf{SE}} (\mathscr{P} \oplus \mathscr{U}) \dot{\vee} (\mathscr{Q} \oplus \mathscr{U})$.

Now we turn to a constructive characterisation of rule update operators satisfying conditions $(\mathrm{P1})_{\mathsf{SE}}$–$(\mathrm{P8})_{\mathsf{SE}}$. Analogically to belief updates, it is based on an order assignment, but this time over the set of all SE-interpretations $\mathcal{X}$. Since the set of SE-models of a program must be well-defined, not every order assignment characterises a rule update operator. We thus additionally define *well-defined order assignments* as those that do.

Definition 17 (*Rule update operator characterised by an order assignment*)
Let $\oplus$ be a rule update operator and ω a preorder assignment over $\mathcal{X}$. We say that $\oplus$ is *characterised by* ω if for all programs $\mathscr{P}$, $\mathscr{U}$,

$$[\![\mathscr{P} \oplus \mathscr{U}]\!]_{\mathsf{SE}} = \bigcup_{X \in [\![\mathscr{P}]\!]_{\mathsf{SE}}} \min\left([\![\mathscr{U}]\!]_{\mathsf{SE}}, \leqslant_{\omega}^{X}\right).$$

We say that a preorder assignment over $\mathcal{X}$ is *well-defined* if some rule update operator is characterised by it.

Similarly, as with belief update, we require the order assignment to be faithful, i.e. to consider each SE-interpretation the closest to itself.

Definition 18 (*Faithful order assignment*)
A preorder assignment ω over $\mathcal{X}$ is *faithful* if for every SE-interpretation X the following condition is satisfied:

$$\text{For every } Y \in \mathcal{X} \text{ with } Y \neq X \text{ it holds that } X <_{\omega}^{X} Y.$$

Interestingly, faithful assignments characterise the same class of operators as the larger class of semi-faithful assignments, defined as follows:

Definition 19 (*Semi-faithful order assignment*)
A preorder assignment ω over $\mathcal{X}$ is *semi-faithful* if for every SE-interpretation X the following conditions are satisfied:

1. For every $Y \in \mathcal{X}$ with $Y \neq X$ and $Y \neq X^*$, either $X <_{\omega}^{X} Y$ or $X^* <_{\omega}^{X} Y$.
2. If $X^* \leqslant_{\omega}^{X} X$, then $X \leqslant_{\omega}^{X} X^*$.

Finally, we require the preorder assignment to satisfy one further condition, related to the well-definedness of sets of SE-models of every program. It can be seen as the natural semantic counterpart of $(\mathrm{P7})_{\mathsf{SE}}$.

Definition 20 (*Organised order assignment*)
A preorder assignment ω is *organised* if for all SE-interpretations X, Y and all well-defined sets of SE-interpretations $\mathscr{M}, \mathscr{N}$ the following condition is satisfied:

If $Y \in \min(\mathscr{M}, \leqslant_{\omega}^{X}) \cup \min(\mathscr{M}, \leqslant_{\omega}^{X^*})$ and $Y \in \min(\mathscr{N}, \leqslant_{\omega}^{X}) \cup \min(\mathscr{N}, \leqslant_{\omega}^{X^*})$,
then $Y \in \min(\mathscr{M} \cup \mathscr{N}, \leqslant_{\omega}^{X}) \cup \min(\mathscr{M} \cup \mathscr{N}, \leqslant_{\omega}^{X^*})$.

Now we are ready to formulate the main result of this section:

Theorem 21 (*Representation theorem for rule updates*)
Let $\oplus$ be a rule update operator. The following conditions are equivalent:

(a) The operator $\oplus$ satisfies conditions $(\mathsf{P1})_{\mathsf{SE}}$–$(\mathsf{P8})_{\mathsf{SE}}$.
(b) The operator $\oplus$ is characterised by a semi-faithful and organised preorder assignment.
(c) The operator $\oplus$ is characterised by a faithful and organised partial order assignment.

Proof
See Appendix A. □

This theorem provides a constructive characterisation of rule update operators satisfying the defined postulates. It facilitates the analysis of their properties, both semantic as well as computational. Note also that it implies that the larger class of *semi-faithful* and organised *preorder* assignments is equivalent to the smaller class of *faithful* and organised *partial order* assignments. Furthermore, it offers a strategy for defining operators satisfying the postulates that can be directly applied whenever an order assignment is known or can be approximated. This strategy is also complete in the sense that, up to strong equivalence, all operators satisfying the postulates can be characterised and distinguished by applying this strategy.

In what follows, we define a specific update operator based on the ideas underlying Winslett's update semantics (Keller and Winslett 1985; Winslett 1990) defined Section 2. Similarly, as was argued in Delgrande *et al.* (2008), since we are working with well-defined sets of SE-interpretations, preference needs to be given to their second component. Thus, we extend the assignment W to all SE-interpretations $X = (I, J)$, $Y = (K_1, L_1)$, $Z = (K_2, L_2)$ as follows: $Y \leqslant_{\mathsf{W}}^{X} Z$ if and only if the following conditions are satisfied:

1. $(L_1 \div J) \subseteq (L_2 \div J)$;
2. If $(L_1 \div J) = (L_2 \div J)$, then $(K_1 \div I) \setminus \Delta \subseteq (K_2 \div I) \setminus \Delta$ where $\Delta = L_1 \div J$.

Intuitively, first we compare the differences between the second components of Y and Z w.r.t. X. If they are equal, we compare the differences between the first components of Y and Z w.r.t. X, but now ignoring the differences between the second components. A concrete illustration of these comparisons is presented next:

Example 22 (*Assignment* W *for* SE*-interpretations*)
Let the SE-interpretations X, Y, Z_1, Z_2, Z_3 be as follows:[1]

$$X = (I, J) = (p, pq), \qquad Y = (K, L) = (p, pr),$$
$$Z_1 = (K_1, L_1) = (p, prs), \quad Z_2 = (K_2, L_2) = (\emptyset, pr), \quad Z_3 = (K_3, L_3) = (pr, pr).$$

We can see that $(L \div J) = \{ q, r \} \subsetneq \{ q, r, s \} = (L_1 \div J)$, so it follows that $Y \leqslant^X_{\mathsf{W}} Z_1$ holds and it is not the case that $Z_1 \leqslant^X_{\mathsf{W}} Y$. Thus, $Y <^X_{\mathsf{W}} Z_1$.

On the other hand, $(L \div J) = (L_2 \div J) = (L_3 \div J) = \Delta = \{ q, r \}$, so Y and Z_2 can only be distinguished based on the second condition. Furthermore, we have $(K \div I) \setminus \Delta = \emptyset \subsetneq \{ p \} = (K_2 \div I) \setminus \Delta$. Similarly as before, we obtain $Y <^X_{\mathsf{W}} Z_2$.

A slightly different case occurs with Z_3 because $(K_3 \div I) \setminus \Delta = \{ r \} \setminus \{ q, r \} = \emptyset$ and it follows that both $Y \leqslant^X_{\mathsf{W}} Z_3$ and $Z_3 \leqslant^X_{\mathsf{W}} Y$ hold, despite the fact that $Y \neq Z_3$.

Our following result shows that W indeed satisfies the necessary conditions to characterise rule update operators satisfying the reformulated postulates.

Proposition 23
The assignment W is a well-defined, faithful and organised preorder assignment.

Proof
See Appendix B. □

Furthermore, as a consequence of Theorem 21 and Proposition 23:

Corollary 24
Every rule update operator characterised by W satisfies conditions $(\mathsf{P1})_{\mathsf{SE}}$–$(\mathsf{P8})_{\mathsf{SE}}$.

As regards the computational complexity of query answering for rule update operators characterised by W, it follows the same pattern as query answering for Winslett's belief update operator (cf. Theorems 6 and 7). In the general case, it resides in the second level of the polynomial hierarchy while for definite programs it drops to the first level. Formally:

Theorem 25 (*Computational complexity of rule updates characterised by* W)
Let $\oplus$ be a rule update operator characterised by W. Deciding whether $\mathscr{P} \oplus \mathscr{U} \models_{\mathsf{SE}} \mathscr{Q}$ for programs $\mathscr{P}$, $\mathscr{U}$, $\mathscr{Q}$ is Π^{P}_2-complete. Hardness holds even if $\mathscr{P}$ is a set of positive facts, $\mathscr{U}$ is a non-disjunctive program and $\mathscr{Q}$ contains a single fact from $\mathscr{P}$.

Proof
See Appendix C. □

Theorem 26 (*Computational complexity of definite rule updates characterised by* W)
Let $\oplus$ be a rule update operator characterised by W. Deciding whether $\mathscr{P} \oplus \mathscr{U} \models_{\mathsf{SE}} \mathscr{Q}$ for definite programs $\mathscr{P}$, $\mathscr{U}$, $\mathscr{Q}$ is co-NP-complete. Hardness holds even if $\mathscr{P}$ is a set of facts and $\mathscr{Q}$ contains a single fact from $\mathscr{P}$.

Proof
See Appendix C. □

[1] For the sake of readability, we omit the usual set notation when listing SE-interpretations. For example, instead of $(\{ p \}, \{ p, q \})$ we simply write (p, pq).

4 Support in semantic rule updates

In this section we take a closer look at the behaviour of semantic rule update operators.

One of the benefits of dealing with rule updates on the semantic level is that semantic properties that are rather difficult to show for syntax-based update operators are much easier to analyse and prove. For example, one of the most widespread and counterintuitive side effects of syntactic updates is that they are sensitive to tautological updates. In case of semantic update operators, such a behaviour is easily shown to be impossible given that the operator satisfies $(\mathsf{P2})_{\mathsf{SE}}$.

However, semantic update operators do not always behave the way we expect. Consider first an example using some update operator $\oplus$ characterised by the order assignment W defined in the previous section:[2]

Example 27
Let the programs $\mathscr{P}$, $\mathscr{Q}$ and $\mathscr{U}$ be as follows:

$$\mathscr{P}: \quad p. \qquad\qquad \mathscr{Q}: \quad p \leftarrow q. \qquad\qquad \mathscr{U}: \quad \sim q.$$
$$\qquad\; q. \qquad\qquad\qquad\quad q.$$

It can be easily verified that:

$$[\![\mathscr{P} \oplus \mathscr{U}]\!]_{\mathsf{SE}} = [\![\mathscr{Q} \oplus \mathscr{U}]\!]_{\mathsf{SE}} = \{\, (p, p) \,\} .$$

Hence, both $\mathscr{P} \oplus \mathscr{U}$ and $\mathscr{Q} \oplus \mathscr{U}$ have the single answer set $J = \{\, p \,\}$. In case of $\mathscr{P} \oplus \mathscr{U}$ this is indeed the expected result. But in case of $\mathscr{Q} \oplus \mathscr{U}$ we can see that p is true in J even though there is no rule in $\mathscr{Q} \cup \mathscr{U}$ justifying it, i.e. there is no rule with p in its head and its body satisfied in J. This means that the behaviour of $\oplus$ is in discord with intuitions underlying most Logic Programming semantics.

In the following, we show that such counterintuitive behaviour is not specific to $\oplus$, but extends to all semantic update operators for answer-set programs based on the well-established notions of SE-models and KM postulates. This is especially interesting from the point of view of comparison with syntax-based approaches to rule updates that, as we formally pinpoint in what follows, do not suffer from such drawbacks.

The property of *support* (Apt *et al.* 1988; Dix 1995b) is one of the basic conditions that Logic Programming semantics are intuitively designed to satisfy. In the static case, this property can be formulated as follows:

[2] It has been shown that Winslett's update semantics has some drawbacks, just as other update operators previously proposed in the context of Classical Logic do (see (Herzig and Rifi 1999 for a survey). Nevertheless, we decided to choose Winslett's update operator as the basis to define a rule update operator and illustrate its properties because it is one of the most extensively studied and understood update operators, and because the undesired behaviour illustrated in this example is shared by all update operators based on KM postulates and SE-models – as we shall see – and not a specific problem due to our choice of Winslett's operator.

Definition 28 (*Static support*)
Let $\mathscr{P}$ be a program, p an atom and J an interpretation. We say that $\mathscr{P}$ *supports* p *in* J if there is some rule $\pi \in \mathscr{P}$ such that $p \in H(\pi)$ and $J \models B(\pi)$.

A Logic Programming semantics S is *supported* if for each model J of a program $\mathscr{P}$ under S the following condition is satisfied: Every atom $p \in J$ is supported by $\mathscr{P}$ in J.

A supported semantics thus requires all atoms in an assigned model to be in the head of some rule with a satisfied body, ensuring that no atom is true without at least *some* justification. Note that the widely accepted Logic Programming semantics, such as the answer-set and well-founded semantics, are supported (see Dix 1995a, 1995b, for more on properties of Logic Programming semantics).

It is only natural to require that rule update operators do not neglect this essential property which also gives rise to much of the intuitive appeal of Logic Programming systems. As it turns out, it is not difficult to verify that despite the substantial differences between various syntax-based approaches to rule updates and revision, all of the semantics introduced in Leite and Pereira (1998), Alferes *et al.* (2000, 2005), Eiter *et al.* (2002), Sakama and Inoue (2003), Zhang (2006), Delgrande *et al.* (2007) and Delgrande (2010) respect support in the following sense:

Definition 29 (*Dynamic support*)
We say that a rule update operator $\oplus$ *respects support* if the following condition is satisfied for all programs $\mathscr{P}$, $\mathscr{U}$ and all answer sets J of $\mathscr{P} \oplus \mathscr{U}$: Every atom $p \in J$ is supported by $\mathscr{P} \cup \mathscr{U}$ in J.

So an update operator respects support if it returns only programs whose answer sets are supported by rules from either the original program or from its update. Similarly as in the case of static support, this amounts to the requirement that an atom may be true only if at least *some* justification can be found for it.

Another basic expectation from an update operator is the usual intuition regarding how facts should be updated by newer facts. It enforces the principle of literal inertia, but only for the case when both the initial program and its update are consistent sets of facts. Similarly as with support, a variety of different syntax-based approaches to rule updates and revision, in particular the semantics introduced in Leite and Pereira (1998), Alferes *et al.* (2000, 2005), Eiter *et al.* (2002), Sakama and Inoue (2003), Zhang (2006), Delgrande *et al.* (2007) and Delgrande (2010), satisfy fact update in the following sense:

Definition 30 (*Fact update*)
We say that a rule update operator $\oplus$ *respects fact update* if for all consistent sets of facts $\mathscr{P}$, $\mathscr{U}$, the unique answer set of $\mathscr{P} \oplus \mathscr{U}$ is the interpretation

$$\{\, p \mid (p.) \in \mathscr{P} \cup \mathscr{U} \land (\sim p.) \notin \mathscr{U} \,\}.$$

Thus, a rule update operator respects fact update if it is well-behaved w.r.t. consistent sets of facts: it provides the answer set that contains exactly those atoms that are asserted as true in either the original program or its update, and are not asserted as false in the update. This behaviour is widely accepted – it stems from the intuitions regarding database updates and is uncontroversial in both the belief change and rule change communities.

We conjecture that any reasonable update operator for answer-set programs should satisfy support and fact update since these two properties place basic constraints on its behaviour and are based on fundamental and widely accepted intuitions. They are by no means exhaustive or sufficient – it is not difficult to define rule update operators that satisfy both of them but are sensitive to tautological updates or quickly end up in an inconsistent state without a possibility of recovery – but they both seem necessary, even elementary, properties of a well-behaved rule update operator. However, it turns out that every rule update operator based on SE-models, even if it satisfies only the basic postulate that enforces syntax independence, fails to comply with at least one of these two basic expectations.

Theorem 31

A rule update operator that satisfies $(\mathsf{P4})_{\mathsf{SE}}$ either does not respect support or it does not respect fact update.

Proof

Let $\oplus$ be a rule update operator that satisfies $(\mathsf{P4})_{\mathsf{SE}}$ and consider again the programs $\mathscr{P}$, $\mathscr{Q}$ and $\mathscr{U}$ from Example 27. Since $\mathscr{P}$ is strongly equivalent to $\mathscr{Q}$, by $(\mathsf{P4})_{\mathsf{SE}}$ we obtain that $\mathscr{P} \oplus \mathscr{U}$ is strongly equivalent to $\mathscr{Q} \oplus \mathscr{U}$. Consequently, $\mathscr{P} \oplus \mathscr{U}$ has the same answer sets as $\mathscr{Q} \oplus \mathscr{U}$. It only remains to observe that if $\oplus$ respects fact update, then $\mathscr{P} \oplus \mathscr{U}$ has the unique answer set $\{p\}$. But then $\{p\}$ is an answer set of $\mathscr{Q} \oplus \mathscr{U}$ in which p is unsupported by $\mathscr{Q} \cup \mathscr{U}$. Hence, $\oplus$ does not respect support. $\square$

So any answer-set program update operator based on SE-models and the KM approach to belief update, as materialised in the fundamental principle $(\mathsf{P4})_{\mathsf{SE}}$, cannot respect two basic and desirable properties: support and fact update. We believe that this is a major drawback of such operators, severely diminishing their applicability.

Moreover, the principle $(\mathsf{P4})_{\mathsf{SE}}$ is also adopted for *revision* of answer-set programs based on SE-models in Delgrande *et al.* (2008).[3] This means that Theorem 31 extends to semantic program revision operators, such as those defined in Delgrande *et al.* (2008): Whenever support and fact update are expected to be satisfied by a rule revision operator, it cannot be defined by purely manipulating the sets of SE-models of the underlying programs.

One question that suggests itself is whether a weaker version of the principle $(\mathsf{P4})_{\mathsf{SE}}$ can be combined with properties such as support and fact update. Its two immediate weakenings, analogous to the weakenings of (B4) in Herzig and Rifi (1999), are as follows:

[3] Note that the belief update postulate (B4), from which $(\mathsf{P4})_{\mathsf{SE}}$ originates, is also one of the reformulated AGM postulates for belief *revision* (Katsuno and Mendelzon 1992). The original AGM framework (Alchourrón *et al.* 1985) assumes that the initial knowledge base $\mathscr{B}$ is closed with respect to logical consequence and the first AGM postulate requires that the result of revision also be a closed set. Under these assumptions, different knowledge bases cannot be equivalent and, as a consequence, the original AGM postulate corresponding to (B4) is $\star 5$: If $Cn(\mu) = Cn(\nu)$, then $\mathscr{B} \star \mu = \mathscr{B} \star \nu$ (where Cn is the logical consequence operator and $\star$ is the revision operator).

$(\mathsf{P4.1})_{\mathsf{SE}}$ If $\mathscr{P} \equiv_{\mathsf{SE}} \mathscr{Q}$, then $\mathscr{P} \oplus \mathscr{U} \equiv_{\mathsf{SE}} \mathscr{Q} \oplus \mathscr{U}$.

$(\mathsf{P4.2})_{\mathsf{SE}}$ If $\mathscr{U} \equiv_{\mathsf{SE}} \mathscr{V}$, then $\mathscr{P} \oplus \mathscr{U} \equiv_{\mathsf{SE}} \mathscr{P} \oplus \mathscr{V}$.

In case of $(\mathsf{P4.1})_{\mathsf{SE}}$, it is easy to see that the proof of Theorem 31 applies in the same way as with $(\mathsf{P4})_{\mathsf{SE}}$, so $(\mathsf{P4.1})_{\mathsf{SE}}$ is likewise incompatible with support and fact update.

On the other hand, principle $(\mathsf{P4.2})_{\mathsf{SE}}$, also referred to as Weak Independence of Syntax (Osorio and Cuevas 2007), does not suffer from such severe limitations. It is, nevertheless, violated by syntax-based rule update semantics that assign a special meaning to occurrences of default literals in heads of rules, as illustrated in the following example:

Example 32
Let the programs $\mathscr{P}$, $\mathscr{U}$ and $\mathscr{V}$ be as follows:

$$\mathscr{P}: \quad p. \qquad\qquad \mathscr{U}: \quad \sim p \leftarrow q. \qquad\qquad \mathscr{V}: \quad \sim q \leftarrow p.$$
$$\quad q.$$

Since $\mathscr{U}$ is strongly equivalent to $\mathscr{V}$, $(\mathsf{P4.2})_{\mathsf{SE}}$ requires that $\mathscr{P} \oplus \mathscr{U}$ be strongly equivalent to $\mathscr{P} \oplus \mathscr{V}$. This is in contrast with the rule update semantics of Leite and Pereira (1998) and Alferes *et al.* (2000, 2005) where a default literal $\sim p$ in the head of a rule indicates that whenever the body of the rule is satisfied, there is a *reason for p to cease being true*. A consequence of this is that an update of $\mathscr{P}$ by $\mathscr{U}$ results in the single answer set $\{ q \}$ while an update by $\mathscr{V}$ leads to the single answer set $\{ p \}$.

Thus, when considering the principle $(\mathsf{P4.2})_{\mathsf{SE}}$, benefits of the declarativeness that it brings with it need to be weighed against the loss of control over the results of updates by rules with default literals in their heads.

The problems we identified might be mitigated if a richer semantic characterisation of logic programs was used instead of SE-models. Such a characterisation would have to be able to distinguish between programs such as $\mathscr{P} = \{ p., q. \}$ and $\mathscr{Q} = \{ p \leftarrow q., q. \}$ because they are expected to behave differently when subject to evolution.

Another alternative is to use one of the syntactic approaches to rule updates, e.g. Alferes *et al.* (2005), that have matured over the years.

5 Conclusion

In this paper, we revisited the problem of updates of answer-set programs, in an attempt to change the focus from the syntactic representation of a program to its semantic content and to facilitate the analysis of semantic properties of defined update operators. We did so by applying the established approach to updates following Katsuno and Mendelzon's postulates in the context of logic programs. Whereas until recently this was not possible since these postulates were simply not applicable (nor adaptable) when considering non-monotonic Logic Programming semantics, as shown in Eiter *et al.* (2002), the introduction of *SE-models* (Turner 2003), which provide a monotonic characterisation of logic programs that is strictly more expressive than the answer-set semantics, provided a new opportunity to cast KM postulates into Logic Programming.

We adapted the KM postulates to be used for answer-set program updates and showed a representation theorem which provides a constructive characterisation of rule update operators satisfying the postulates. This characterisation not only facilitates the investigation of these operators' properties, both semantic as well as computational, but it also provides an intuitive strategy for constructively defining these operators. This is one of the major contributions of the paper since it brings, for the first time, updates of answer-set programs in line with KM postulates. We illustrated this result with a definition of a specific rule update operator which is a counterpart of Winslett's belief update operator.

The second important contribution of this paper is the uncovering of a serious drawback that extends to all answer-set program update operators based on SE-models and AGM-style approach to program revision and update. All such operators violate at least one of two basic and very desirable properties. The first one consists of respecting *support*, a property that is enjoyed, in the static case, by all widely accepted Logic Programming semantics. The second property, *fact update*, is concerned with the answer set assigned to a consistent set of facts after it is updated by another consistent set of facts. This contribution is very important as it should guide further research on updates of answer-set programs

(a) away from the purely semantic approach materialised in AGM and KM postulates, or
(b) to the development of semantic characterisations of answer-set programs that are richer than SE-models and appropriately capture their dynamic behaviour, such as in Slota and Leite (2012), or even
(c) turning back to the more syntactic approaches, such as Alferes *et al.* (2005), and see whether they indeed offer a viable alternative.

Either way, updating answer-set programs is a very important theoretical and practical problem that is still waiting for a definite solution. Also, despite the issues with the syntax independence postulate $(\mathsf{P4})_{\mathsf{SE}}$, other principles based on SE-models play an important role with regards to the classification and evaluation of different approaches to rule change. For instance, the reformulations of rule change principles from Eiter *et al.* (2002) in terms of strong equivalence, considered already in Delgrande *et al.* (2008), can be formulated as follows:

$(\mathsf{Initialisation})_{\mathsf{SE}}$ $\emptyset \oplus \mathscr{U} \equiv_{\mathsf{SE}} \mathscr{U}$.

$(\mathsf{Idempotence})_{\mathsf{SE}}$ $\mathscr{P} \oplus \mathscr{P} \equiv_{\mathsf{SE}} \mathscr{P}$.

$(\mathsf{Tautology})_{\mathsf{SE}}$ If $\mathscr{U} \equiv_{\mathsf{SE}} \emptyset$, then $\mathscr{P} \oplus \mathscr{U} \equiv_{\mathsf{SE}} \mathscr{P}$.

$(\mathsf{Absorption})_{\mathsf{SE}}$ If $\mathscr{U} \equiv_{\mathsf{SE}} \mathscr{V}$, then $(\mathscr{P} \oplus \mathscr{U}) \oplus \mathscr{V} \equiv_{\mathsf{SE}} \mathscr{P} \oplus \mathscr{V}$.

$(\mathsf{Augmentation})_{\mathsf{SE}}$ If $\mathscr{V} \models_{\mathsf{SE}} \mathscr{U}$, then $(\mathscr{P} \oplus \mathscr{U}) \oplus \mathscr{V} \equiv_{\mathsf{SE}} \mathscr{P} \oplus \mathscr{V}$.

We believe that all of these properties are indeed desirable and strengthen their original formulations in an interesting way. Investigation of operators with these properties, as well as a further analysis of the postulates $(\mathsf{P1})_{\mathsf{SE}}$–$(\mathsf{P8})_{\mathsf{SE}}$, remains an important research topic. This paper contains, we believe, a relevant contribution to a better understanding of rule change that will help guide future research.

Acknowledgement

We thank the anonymous reviewers for their valuable comments. M. Slota was supported by FCT scholarship SFRH/BD/38214/2007. J. Leite was partially supported by FCT funded project ERRO (PTDC/EIA-CCO/121823/2010).

Appendix A Proofs: representation theorem

Definition 33 (*Program corresponding to a set of* SE*-interpretations*)
Let $\mathscr{M}$ be a set of SE-interpretations. We denote by $\|\mathscr{M}\|$ some arbitrary but fixed program $\mathscr{P}$ such that

$$[\![\mathscr{P}]\!]_{\mathsf{SE}} = \{\, X, X^* \mid X \in \mathscr{M} \,\}\,.$$

Instead of $\|\{ X_1, X_2, \dots, X_n \}\|$ we usually write $\|X_1, X_2, \dots, X_n\|$.

Definition 34 (*Order assignment generated by an update operator*)
Let $\oplus$ be a rule update operator and X an SE-interpretation. We define the binary relation $\prec_\oplus^X$ for all SE-interpretations Y, Z as follows: $Y \prec_\oplus^X Z$ if and only if the following conditions are satisfied:

$$Y \in [\![\, \|X\| \oplus \|Y, Z\| \,]\!]_{\mathsf{SE}} \tag{A1}$$

$$Z \notin [\![\, \|X\| \oplus \|Y, Z\| \,]\!]_{\mathsf{SE}} \tag{A2}$$

$$\text{If } Y \neq Y^*, \text{ then } Z \in [\![\, \|X\| \oplus \|Y^*, Z\| \,]\!]_{\mathsf{SE}} \tag{A3}$$

The *preorder assignment generated by* $\oplus$ assigns to every SE-interpretation X the reflexive and transitive closure $\leqslant_\oplus^X$ of $\prec_\oplus^X$, i.e. $Y \leqslant_\oplus^X Z$ if and only if $Y = Z$ or there is some $n \geqslant 2$ and SE-interpretations $Y_1, Y_2, \dots, Y_n$ such that $Y = Y_1 \prec_\oplus^X Y_2 \prec_\oplus^X \cdots \prec_\oplus^X Y_n = Z$.

Lemma 35
Let $\oplus$ be a rule update operator satisfying conditions $(\mathsf{P1})_{\mathsf{SE}}$–$(\mathsf{P8})_{\mathsf{SE}}$ and X, Y, Z some SE-interpretations. If $Y \leqslant_\oplus^X Z$, then either $Y = Z$ or $Z \notin [\![\, \|X\| \oplus \|Y, Z\| \,]\!]_{\mathsf{SE}}$.

Proof
Suppose that $Y \neq Z$. Then, by the definition of $\leqslant_\oplus^X$, for some $n \leqslant 2$ and SE-interpretations $Y_1, Y_2, \dots, Y_n$ it holds that $Y = Y_1 \prec_\oplus^X Y_2 \prec_\oplus^X \cdots \prec_\oplus^X Y_n = Z$. We will prove by induction on n that $Y_n \notin [\![\, \|X\| \oplus \|Y_1, Y_n\| \,]\!]_{\mathsf{SE}}$ from which the desired result follows directly.

1° For $n = 2$ this follows from $Y_1 \prec_\oplus^X Y_2$ by (A2).
2° We inductively assume that

$$Y_n \notin [\![\, \|X\| \oplus \|Y_1, Y_n\| \,]\!]_{\mathsf{SE}} \tag{A4}$$

and prove that $Y_{n+1} \notin [\![\, \|X\| \oplus \|Y_1, Y_{n+1}\| \,]\!]_{\mathsf{SE}}$.
We know that $Y_n \prec_\oplus^X Y_{n+1}$, so by (A2) we obtain

$$Y_{n+1} \notin [\![\, \|X\| \oplus \|Y_n, Y_{n+1}\| \,]\!]_{\mathsf{SE}}. \tag{A5}$$

Considering that the program $\|Y_1, Y_n, Y_{n+1}\| \dot{\wedge} \|Y_1, Y_n\|$ is strongly equivalent to $\|Y_1, Y_n\|$, by $(\mathsf{P5})_{\mathsf{SE}}$ and $(\mathsf{P4})_{\mathsf{SE}}$ we conclude that

$$(\|X\| \oplus \|Y_1, Y_n, Y_{n+1}\|) \dot{\wedge} \|Y_1, Y_n\| \models_{\mathsf{SE}} \|X\| \oplus \|Y_1, Y_n\|$$

which, together with (A4), implies that

$$Y_n \notin [\![\|X\| \oplus \|Y_1, Y_n, Y_{n+1}\|]\!]_{\mathsf{SE}}. \tag{A6}$$

Similarly, since the program $\|Y_1, Y_n, Y_{n+1}\| \dot{\wedge} \|Y_n, Y_{n+1}\|$ is strongly equivalent to $\|Y_n, Y_{n+1}\|$, by $(\mathsf{P5})_{\mathsf{SE}}$ and $(\mathsf{P4})_{\mathsf{SE}}$ we obtain that

$$(\|X\| \oplus \|Y_1, Y_n, Y_{n+1}\|) \dot{\wedge} \|Y_n, Y_{n+1}\| \models_{\mathsf{SE}} \|X\| \oplus \|Y_n, Y_{n+1}\|,$$

and so due to (A5) it holds that

$$Y_{n+1} \notin [\![\|X\| \oplus \|Y_1, Y_n, Y_{n+1}\|]\!]_{\mathsf{SE}}. \tag{A7}$$

Now we consider two cases:

(a) If $Y_n = Y_n^*$, then (A6) and $(\mathsf{P1})_{\mathsf{SE}}$ imply that

$$\|X\| \oplus \|Y_1, Y_n, Y_{n+1}\| \models_{\mathsf{SE}} \|Y_1, Y_{n+1}\|;$$
$$\|X\| \oplus \|Y_1, Y_{n+1}\| \models_{\mathsf{SE}} \|Y_1, Y_n, Y_{n+1}\|,$$

so by $(\mathsf{P6})_{\mathsf{SE}}$ we can conclude that $\|X\| \oplus \|Y_1, Y_n, Y_{n+1}\|$ is strongly equivalent to $\|X\| \oplus \|Y_1, Y_{n+1}\|$. But then the desired conclusion follows from (A7).

(b) If $Y_n \neq Y_n^*$, then from (A3) we infer that

$$Y_{n+1} \in [\![\|X\| \oplus \|Y_n^*, Y_{n+1}\|]\!]_{\mathsf{SE}}. \tag{A8}$$

Furthermore, from (A6) and $(\mathsf{P1})_{\mathsf{SE}}$ we obtain

$$\|X\| \oplus \|Y_1, Y_n, Y_{n+1}\| \models_{\mathsf{SE}} \|Y_1, Y_n^*, Y_{n+1}\|;$$
$$\|X\| \oplus \|Y_1, Y_n^*, Y_{n+1}\| \models_{\mathsf{SE}} \|Y_1, Y_n, Y_{n+1}\|,$$

so by $(\mathsf{P6})_{\mathsf{SE}}$ we can conclude that $\|X\| \oplus \|Y_1, Y_n, Y_{n+1}\|$ is strongly equivalent to $\|X\| \oplus \|Y_1, Y_n^*, Y_{n+1}\|$ and, due to (A7),

$$Y_{n+1} \notin [\![\|X\| \oplus \|Y_1, Y_n^*, Y_{n+1}\|]\!]_{\mathsf{SE}}.$$

Since $\|Y_1, Y_n^*, Y_{n+1}\|$ is strongly equivalent to $\|Y_1, Y_{n+1}\| \dot{\vee} \|Y_n^*, Y_{n+1}\|$, it follows from $(\mathsf{P4})_{\mathsf{SE}}$ and $(\mathsf{P7})_{\mathsf{SE}}$ that either $Y_{n+1} \notin [\![\|X\| \oplus \|Y_1, Y_{n+1}\|]\!]_{\mathsf{SE}}$ or $Y_{n+1} \notin [\![\|X\| \oplus \|Y_n^*, Y_{n+1}\|]\!]_{\mathsf{SE}}$. The latter is impossible due to (A8). □

Lemma 36

Let $\oplus$ be a rule update operator satisfying conditions $(\mathsf{P1})_{\mathsf{SE}}$–$(\mathsf{P8})_{\mathsf{SE}}$ and X, Y, Z, some SE-interpretations. If $Y \not\leq_{\oplus}^{X} Z$, then the following conditions are satisfied:

(1) If $Y = Z^*$, then $Z \in [\![\|X\| \oplus \|Y, Z\|]\!]_{\mathsf{SE}}$.
(2) If $Y = Y^*$ and $Z \in [\![\|X\| \oplus \|Z\|]\!]_{\mathsf{SE}}$, then $Z \in [\![\|X\| \oplus \|Y, Z\|]\!]_{\mathsf{SE}}$.
(3) If $Y \neq Y^*$ and $Z \in [\![\|X\| \oplus \|Y^*, Z\|]\!]_{\mathsf{SE}}$, then $Z \in [\![\|X\| \oplus \|Y, Z\|]\!]_{\mathsf{SE}}$.

Proof
First, we show the following auxiliary statement: If $Y = Z$ or $Y \notin [\![\|X\| \oplus \|Y,Z\|]\!]_{\mathsf{SE}}$, then all three conditions are satisfied.

First, suppose that $Y = Z$. If $Y = Z^*$, then $Y = Y^* = Z = Z^*$, so it follows from $(\mathsf{P1})_{\mathsf{SE}}$ and $(\mathsf{P3})_{\mathsf{SE}}$ that $[\![\|X\| \oplus \|Y,Z\|]\!]_{\mathsf{SE}} = [\![\|X\| \oplus \|Z^*\|]\!]_{\mathsf{SE}} = \{ Z^* \}$, verifying condition (1). Furthermore, conditions (2) and (3) are satisfied because $\|Z\| = \|Y^*,Z\| = \|Y,Z\|$.

Now suppose that $Y \notin [\![\|X\| \oplus \|Y,Z\|]\!]_{\mathsf{SE}}$. If $Y = Z^*$, then it follows from $(\mathsf{P1})_{\mathsf{SE}}$ and $(\mathsf{P3})_{\mathsf{SE}}$ that $Z \in [\![\|X\| \oplus \|Y,Z\|]\!]_{\mathsf{SE}}$. If $Y = Y^*$, then it follows from $(\mathsf{P1})_{\mathsf{SE}}$ that

$$\|X\| \oplus \|Y,Z\| \models_{\mathsf{SE}} \|Z\| \qquad \text{and} \qquad \|X\| \oplus \|Z\| \models_{\mathsf{SE}} \|Y,Z\|,$$

so by $(\mathsf{P6})_{\mathsf{SE}}$ we obtain that $\|X\| \oplus \|Y,Z\| \equiv_{\mathsf{SE}} \|X\| \oplus \|Z\|$. Hence, it follows from $Z \in [\![\|X\| \oplus \|Z\|]\!]_{\mathsf{SE}}$ that $Z \in [\![\|X\| \oplus \|Y,Z\|]\!]_{\mathsf{SE}}$. On the other hand, if $Y \neq Y^*$, then it follows from $(\mathsf{P1})_{\mathsf{SE}}$ that

$$\|X\| \oplus \|Y,Z\| \models_{\mathsf{SE}} \|Y^*,Z\| \qquad \text{and} \qquad \|X\| \oplus \|Y^*,Z\| \models_{\mathsf{SE}} \|Y,Z\|,$$

so by $(\mathsf{P6})_{\mathsf{SE}}$ we obtain that $\|X\| \oplus \|Y,Z\| \equiv_{\mathsf{SE}} \|X\| \oplus \|Y^*,Z\|$. Hence it follows from $Z \in [\![\|X\| \oplus \|Y^*,Z\|]\!]_{\mathsf{SE}}$ that $Z \in [\![\|X\| \oplus \|Y,Z\|]\!]_{\mathsf{SE}}$.

Turning to the proof of the lemma, note that since $Y \not<^X_\oplus Z$, either $Y \not\leqslant^X_\oplus Z$ or $Z \leqslant^X_\oplus Y$. In the former case, $Y \not<^X_\oplus Z$, so, by the definition of $<^X_\oplus$, either $Y \notin [\![\|X\| \oplus \|Y,Z\|]\!]_{\mathsf{SE}}$, so we can apply our auxiliary statement, or $Z \in [\![\|X\| \oplus \|Y,Z\|]\!]_{\mathsf{SE}}$ as desired, or $Y \neq Y^*$ and $Z \notin [\![\|X\| \oplus \|Y^*,Z\|]\!]_{\mathsf{SE}}$, in which case all three conditions are trivially satisfied. In the latter case it follows from Lemma 35 that either $Y = Z$ or $Y \notin [\![\|X\| \oplus \|Y,Z\|]\!]_{\mathsf{SE}}$, so the rest follows once again from the auxiliary statement. $\square$

Proposition 37
Let $\oplus$ be a rule update operator satisfying conditions $(\mathsf{P1})_{\mathsf{SE}}$–$(\mathsf{P8})_{\mathsf{SE}}$, X an SE-interpretation and $\mathcal{U}$ a program. Then,

$$[\![\|X\| \oplus \mathcal{U}]\!]_{\mathsf{SE}} = \min \left([\![\mathcal{U}]\!]_{\mathsf{SE}}, \leqslant^X_\oplus \right).$$

Proof
First take some $Z \in [\![\|X\| \oplus \mathcal{U}]\!]_{\mathsf{SE}}$. By $(\mathsf{P1})_{\mathsf{SE}}$, $Z \in [\![\mathcal{U}]\!]_{\mathsf{SE}}$. Suppose that Z is not minimal in $[\![\mathcal{U}]\!]_{\mathsf{SE}}$ w.r.t. $\leqslant^X_\oplus$. Then there is some $Y \in [\![\mathcal{U}]\!]_{\mathsf{SE}}$ such that $Y <^X_\oplus Z$. Thus, $Y \neq Z$, and by Lemma 35 we conclude that $Z \notin [\![\|X\| \oplus \|Y,Z\|]\!]_{\mathsf{SE}}$. Considering that $\mathcal{U} \dot\wedge \|Y,Z\|$ is strongly equivalent to $\|Y,Z\|$, it follows from $(\mathsf{P4})_{\mathsf{SE}}$ and $(\mathsf{P5})_{\mathsf{SE}}$ that $(\|X\| \oplus \mathcal{U}) \dot\wedge \|Y,Z\| \models_{\mathsf{SE}} \|X\| \oplus \|Y,Z\|$. Consequently, $Z \notin [\![\|X\| \oplus \mathcal{U}]\!]_{\mathsf{SE}}$, contrary to our assumption. Therefore, $[\![\|X\| \oplus \mathcal{U}]\!]_{\mathsf{SE}}$ is a subset of $\min([\![\mathcal{U}]\!]_{\mathsf{SE}}, \leqslant^X_\oplus)$.

To prove the converse inclusion, assume that Z is minimal in $[\![\mathcal{U}]\!]_{\mathsf{SE}}$ w.r.t. $\leqslant^X_\oplus$ and take some $Y \in [\![\mathcal{U}]\!]_{\mathsf{SE}}$. Note that $Y \not<^X_\oplus Z$, so we can use Lemma 36. We will show that $Z \in [\![\|X\| \oplus \|Y,Z\|]\!]_{\mathsf{SE}}$. We consider three cases:

(a) If $Y = Z^*$, then $Z \in [\![\|X\| \oplus \|Y,Z\|]\!]_{\mathsf{SE}}$ follows immediately from condition (1) of Lemma 36.

(b) If $Y = Y^*$, then the previous case together with the fact that $[\![\mathscr{U}]\!]_{\mathsf{SE}}$ is well-defined entails that $Z \in [\![\, \|X\| \oplus \|Z\| \,]\!]_{\mathsf{SE}}$ and by condition (2) of Lemma 36 it follows that $Z \in [\![\, \|X\| \oplus \|Y, Z\| \,]\!]_{\mathsf{SE}}$.
(c) If $Y \neq Y^*$, then the previous case together with the fact that $[\![\mathscr{U}]\!]_{\mathsf{SE}}$ is well-defined entails that $Z \in [\![\, \|X\| \oplus \|Y^*, Z\| \,]\!]_{\mathsf{SE}}$ and by condition (3) of Lemma 36 it follows that $Z \in [\![\, \|X\| \oplus \|Y, Z\| \,]\!]_{\mathsf{SE}}$.

The choice of Y was arbitrary, so we have proven that $Z \in [\![\, \|X\| \oplus \|Y, Z\| \,]\!]_{\mathsf{SE}}$ for all $Y \in [\![\mathscr{U}]\!]_{\mathsf{SE}}$. This means that by repeated application of $(\mathsf{P7})_{\mathsf{SE}}$, Z is an SE-model of the program

$$\|X\| \oplus \dot{\bigvee_{Y \in [\![\mathscr{U}]\!]_{\mathsf{SE}}}} \|Y, Z\|$$

and since $\mathscr{U}$ is strongly equivalent to the program $\dot{\bigvee}_{Y \in [\![\mathscr{U}]\!]_{\mathsf{SE}}} \|Y, Z\|$, it follows from $(\mathsf{P4})_{\mathsf{SE}}$ that $Z \in [\![\, \|X\| \oplus \mathscr{U} \,]\!]_{\mathsf{SE}}$. $\square$

Proposition 38
If a rule update operator $\oplus$ satisfies conditions $(\mathsf{P1})_{\mathsf{SE}}$–$(\mathsf{P8})_{\mathsf{SE}}$, then the preorder assignment generated by $\oplus$ is semi-faithful and organised and it characterises $\oplus$.

Proof
First we show that the assignment generated by $\oplus$ characterises $\oplus$. We know that $\mathscr{P}$ is strongly equivalent to the program $\dot{\bigvee}_{X \in [\![\mathscr{P}]\!]_{\mathsf{SE}}} \|X\|$, so by $(\mathsf{P4})_{\mathsf{SE}}$ and repeated application of $(\mathsf{P8})_{\mathsf{SE}}$ we obtain that $\mathscr{P} \oplus \mathscr{U}$ is strongly equivalent to the program

$$\dot{\bigvee_{X \in [\![\mathscr{P}]\!]_{\mathsf{SE}}}} (\|X\| \oplus \mathscr{U}).$$

Furthermore, Proposition 37 implies that $[\![\, \|X\| \oplus \mathscr{U} \,]\!]_{\mathsf{SE}} = \min\left([\![\mathscr{U}]\!]_{\mathsf{SE}}, \leqslant_{\oplus}^{X}\right)$, so indeed

$$[\![\mathscr{P} \oplus \mathscr{U}]\!]_{\mathsf{SE}} = \bigcup_{X \in [\![\mathscr{P}]\!]_{\mathsf{SE}}} [\![\, \|X\| \oplus \mathscr{U} \,]\!]_{\mathsf{SE}} = \bigcup_{X \in [\![\mathscr{P}]\!]_{\mathsf{SE}}} \min\left([\![\mathscr{U}]\!]_{\mathsf{SE}}, \leqslant_{\oplus}^{X}\right). \tag{A9}$$

To see that the assignment generated by $\oplus$ is semi-faithful, first take some SE-interpretations X, Y such that $Y \neq X$ and $Y \neq X^*$. We need to show that either $X <_{\oplus}^{X} Y$ or $X^* <_{\oplus}^{X} Y$. Equation (A9) together with $(\mathsf{P2})_{\mathsf{SE}}$ implies that

$$\begin{aligned}
[\![\, \|X\| \oplus \|Y^*, X\| \,]\!]_{\mathsf{SE}} &= \min\left(\{\, Y^*, X, X^* \,\}, \leqslant_{\oplus}^{X}\right) \cup \min\left(\{\, Y^*, X, X^* \,\}, \leqslant_{\oplus}^{X^*}\right) \\
&= \{\, X, X^* \,\}, \\
[\![\, \|X\| \oplus \|Y, X\| \,]\!]_{\mathsf{SE}} &= \min\left(\{\, Y, Y^*, X, X^* \,\}, \leqslant_{\oplus}^{X}\right) \cup \min\left(\{\, Y, Y^*, X, X^* \,\}, \leqslant_{\oplus}^{X^*}\right) \\
&= \{\, X, X^* \,\}.
\end{aligned}$$

Thus, Y^* is not minimal within $\{\, Y^*, X, X^* \,\}$ and Y is not minimal within $\{\, Y, Y^*, X, X^* \,\}$ w.r.t. $\leqslant_{\oplus}^{X}$. In other words:

$$\text{either } X <_{\oplus}^{X} Y^* \text{ or } X^* <_{\oplus}^{X} Y^* \text{ and} \tag{A10}$$

$$\text{either } X <_{\oplus}^{X} Y \text{ or } X^* <_{\oplus}^{X} Y \text{ or } Y^* <_{\oplus}^{X} Y. \tag{A11}$$

In case of the first two alternatives of (A11), we have already achieved our goal. The third alternative together with (A10) and transitivity of $<^X_\oplus$ also concludes the proof of the first condition of semi-faithfulness. To see that the second condition holds as well, consider that by $(\mathsf{P2})_{\mathsf{SE}}$, $[\![\|X^*\| \oplus \|X\|]\!]_{\mathsf{SE}} = \{ X^* \}$ and $[\![\|X\| \oplus \|X\|]\!]_{\mathsf{SE}} = \{ X, X^* \}$, so it follows from (A9) that

$$X \notin \min(\{ X, X^* \}, \leqslant^{X^*}_\oplus) \quad \text{and} \quad X \in \min(\{ X, X^* \}, \leqslant^{X}_\oplus) \cup \min(\{ X, X^* \}, \leqslant^{X^*}_\oplus).$$

Hence, $X \in \min(\{ X, X^* \}, \leqslant^X_\oplus)$. In other words, if $X^* \leqslant^X_\oplus X$, then it must also be the case that $X \leqslant^X_\oplus X^*$. Consequently, the order assignment generated by $\oplus$ is semi-faithful.

To show that it is also organised, consider well-defined sets of SE-interpretations $\mathscr{M}$, $\mathscr{N}$, and SE-interpretations X, Y such that

$$Y \in \min\left(\mathscr{M}, \leqslant^X_\oplus\right) \cup \min\left(\mathscr{M}, \leqslant^{X^*}_\oplus\right) \quad \text{and} \quad Y \in \min\left(\mathscr{N}, \leqslant^X_\oplus\right) \cup \min\left(\mathscr{N}, \leqslant^{X^*}_\oplus\right).$$

By (A9) we obtain that $Y \in [\![\|X\| \oplus \|\mathscr{M}\|]\!]_{\mathsf{SE}}$ and $Y \in [\![\|X\| \oplus \|\mathscr{N}\|]\!]_{\mathsf{SE}}$. Applying $(\mathsf{P7})_{\mathsf{SE}}$ and $(\mathsf{P4})_{\mathsf{SE}}$ yields that $Y \in [\![\|X\| \oplus \|\mathscr{M} \cup \mathscr{N}\|]\!]_{\mathsf{SE}}$. Consequently, by (A9), either $Y \in \min\left(\mathscr{M} \cup \mathscr{N}, \leqslant^X_\oplus\right)$ or $Y \in \min\left(\mathscr{M} \cup \mathscr{N}, \leqslant^{X^*}_\oplus\right)$, so the order assignment generated by $\oplus$ is organised. $\square$

Lemma 39
Let ω be a semi-faithful preorder assignment and X an SE-interpretation. Then there is no SE-interpretation Y such that $Y <^X_\omega X$.

Proof
We prove by contradiction. Suppose that $Y <^X_\omega X$ for some SE-interpretation Y. Clearly, $Y \neq X$ due to irreflexivity of $<^X_\omega$ and $Y \neq X^*$ due to the second condition of semi-faithfulness. Hence, $Y \neq X$ and $Y \neq X^*$, so by the first condition of semi-faithfulness, either $X <^X_\omega Y$ or $X^* <^X_\omega Y$. The former is in conflict with the irreflexivity of $<^X_\omega$ and in the latter case it follows by transitivity of $<^X_\omega$ that $X^* <^X_\omega X$, contrary to the second condition of semi-faithfulness. $\square$

Proposition 40
Let $\oplus$ be a rule update operator. If $\oplus$ is characterised by a semi-faithful and organised preorder assignment, then it is also characterised by a faithful and organised partial order assignment.

Proof
Let $\oplus$ be characterised by a semi-faithful and organised preorder assignment ω. We define the assignment ω' over $\mathfrak{X}$ as follows:

$$Y \leqslant^X_{\omega'} Z \quad \text{if and only if} \quad Y = X \vee Y = Z \vee Y <^X_\omega Z.$$

We need to show that $\leqslant^X_{\omega'}$ is a partial order for all $X \in \mathfrak{X}$, that ω' is faithful and organised and that for all programs $\mathscr{P}$, $\mathscr{U}$,

$$[\![\mathscr{P} \oplus \mathscr{U}]\!]_{\mathsf{SE}} = \bigcup_{X \in [\![\mathscr{P}]\!]_{\mathsf{SE}}} \min\left([\![\mathscr{U}]\!]_{\mathsf{SE}}, \leqslant^X_{\omega'}\right).$$

Note that due to Lemma 39, the following holds for all SE-interpretations X, Y:

$$\text{If } Y \leqslant^X_{\omega'} X, \text{ then } Y = X. \tag{A12}$$

Otherwise we would obtain that $Y <^X_\omega X$, which is in conflict with Lemma 39.

Turning back to the main proof, reflexivity of $\leqslant^X_{\omega'}$ follows directly by its definition.

To show that $\leqslant^X_{\omega'}$ is antisymmetric, take some SE-interpretations Y_1, Y_2 such that $Y_1 \leqslant^X_{\omega'} Y_2$ and $Y_2 \leqslant^X_{\omega'} Y_1$. If $Y_1 = X$, then $Y_2 \leqslant^X_{\omega'} X$ and it follows from (A12) that $Y_2 = X = Y_1$. The case when $Y_2 = X$ is symmetric. If $Y_1 \neq X$ and $Y_2 \neq X$, then, by the definition of $\leqslant^X_{\omega'}$, either $Y_1 = Y_2$ as desired, or $Y_1 <^X_\omega Y_2$ and $Y_2 <^X_\omega Y_1$, which is in conflict with the transitivity and irreflexivity of $<^X_\omega$.

Turning to transitivity of $\leqslant^X_{\omega'}$, suppose that $Y_1 \leqslant^X_{\omega'} Y_2$ and $Y_2 \leqslant^X_{\omega'} Y_3$. We need to show that $Y_1 \leqslant^X_{\omega'} Y_3$. We consider three cases:

(a) If $Y_1 = X$, then $Y_1 \leqslant^X_{\omega'} Y_3$ by the definition of $\leqslant^X_{\omega'}$.
(b) If $Y_2 = X$, then $Y_1 \leqslant^X_{\omega'} X$, so $Y_1 = X$ due to (A12) and the previous case applies.
(c) If $Y_1 \neq X$ and $Y_2 \neq X$, then the desired conclusion follows from the transitivity of equality and of $<^X_\omega$.

As for faithfulness of ω', suppose that $Y \neq X$. We have $X \leqslant^X_{\omega'} Y$ by definition and $Y \not\leqslant^X_{\omega'} X$ follows from (A12).

To show that ω' is organised, we prove the following property: For any well-defined set of SE-interpretations $\mathscr{M}$ and any SE-interpretation X,

$$\min\left(\mathscr{M}, \leqslant^X_{\omega'}\right) \cup \min\left(\mathscr{M}, \leqslant^{X^*}_{\omega'}\right) = \min\left(\mathscr{M}, \leqslant^X_{\omega}\right) \cup \min\left(\mathscr{M}, \leqslant^{X^*}_{\omega}\right). \tag{A13}$$

From (A13) it follows that since ω is organised, ω' must also be.

Before we prove (A13), we need to note that $Y <^X_{\omega'} Z$ holds if and only if $Y \leqslant^X_{\omega'} Z$ and $Z \not\leqslant^X_{\omega'} Y$, so according to the definition of $\leqslant^X_{\omega'}$,

$$Y <^X_{\omega'} Z \quad \text{if and only if} \quad (Y = X \vee Y = Z \vee Y <^X_\omega Z) \wedge (Z \neq X \wedge Z \neq Y \wedge Z \not<^X_\omega Y).$$

Due to Lemma 39 and the transitivity and irreflexivity of $<^X_\omega$, this can be simplified to

$$Y <^X_{\omega'} Z \quad \text{if and only if} \quad (Y = X \wedge Y \neq Z) \vee Y <^X_\omega Z. \tag{A14}$$

Coming back to the proof of (A13), we need to consider three cases:

(a) If $X \notin \mathscr{M}$ and $X^* \notin \mathscr{M}$, then for all $Y, Z \in \mathscr{M}$, $Y \neq X$ and $Y \neq X^*$, so by (A14),

$$Y <^X_{\omega'} Z \text{ if and only if } Y <^X_\omega Z \quad \text{and} \quad Y <^{X^*}_{\omega'} Z \text{ if and only if } Y <^{X^*}_\omega Z,$$

from which the desired conclusion follows directly.
(b) If $X \notin \mathscr{M}$ and $X^* \in \mathscr{M}$, then for all $Y, Z \in \mathscr{M}$, $Y \neq X$, so by (A14),

$$Y <^X_{\omega'} Z \text{ if and only if } Y <^X_\omega Z.$$

Consequently, $\min(\mathscr{M}, \leqslant^X_{\omega'}) = \min(\mathscr{M}, \leqslant^X_\omega)$, and by (A14) and semi-faithfulness of ω we obtain $\min(\mathscr{M}, \leqslant^{X^*}_{\omega'}) = \{ X^* \} = \min(\mathscr{M}, \leqslant^{X^*}_\omega)$.

(c) If $X \in \mathscr{M}$, then $X^* \in \mathscr{M}$, and by (A14) and semi-faithfulness of ω,

$$\{X\} \subseteq \min\left(\mathscr{M}, \leqslant_\omega^X\right) \subseteq \{X, X^*\}, \qquad \min\left(\mathscr{M}, \leqslant_\omega^{X^*}\right) = \{X^*\},$$
$$\min\left(\mathscr{M}, \leqslant_{\omega'}^X\right) = \{X\}, \qquad \min\left(\mathscr{M}, \leqslant_{\omega'}^{X^*}\right) = \{X^*\},$$

from which the desired conclusion follows straightforwardly.

Finally, it follows from the assumption that ω characterises $\oplus$ and from (A13) that

$$\begin{aligned}
[\![\mathscr{P} \oplus \mathscr{U}]\!]_{\mathsf{SE}} &= \bigcup_{X \in [\![\mathscr{P}]\!]_{\mathsf{SE}}} \min\left([\![\mathscr{U}]\!]_{\mathsf{SE}}, \leqslant_\omega^X\right) \\
&= \bigcup_{X \in [\![\mathscr{P}]\!]_{\mathsf{SE}}} \left(\min\left([\![\mathscr{U}]\!]_{\mathsf{SE}}, \leqslant_\omega^X\right) \cup \min\left([\![\mathscr{U}]\!]_{\mathsf{SE}}, \leqslant_\omega^{X^*}\right)\right) \\
&= \bigcup_{X \in [\![\mathscr{P}]\!]_{\mathsf{SE}}} \left(\min\left([\![\mathscr{U}]\!]_{\mathsf{SE}}, \leqslant_{\omega'}^X\right) \cup \min\left([\![\mathscr{U}]\!]_{\mathsf{SE}}, \leqslant_{\omega'}^{X^*}\right)\right) \\
&= \bigcup_{X \in [\![\mathscr{P}]\!]_{\mathsf{SE}}} \min\left([\![\mathscr{U}]\!]_{\mathsf{SE}}, \leqslant_{\omega'}^X\right). \quad \square
\end{aligned}$$

Proposition 41

Let $\oplus$ be a rule update operator. If $\oplus$ is characterised by a faithful and organised partial order assignment, then $\oplus$ satisfies conditions $(\mathsf{P1})_{\mathsf{SE}}$–$(\mathsf{P8})_{\mathsf{SE}}$.

Proof

Let $\oplus$ be characterised by a faithful and organised partial order assignment ω. We consider each condition separately:

$(\mathsf{P1})_{\mathsf{SE}}$ Since ω characterises $\oplus$, for all programs $\mathscr{P}$, $\mathscr{U}$,

$$[\![\mathscr{P} \oplus \mathscr{U}]\!]_{\mathsf{SE}} = \bigcup_{X \in [\![\mathscr{P}]\!]_{\mathsf{SE}}} \min\left([\![\mathscr{U}]\!]_{\mathsf{SE}}, \leqslant_\omega^X\right),$$

so all elements of $[\![\mathscr{P} \oplus \mathscr{U}]\!]_{\mathsf{SE}}$ belong to $[\![\mathscr{U}]\!]_{\mathsf{SE}}$. Equivalently, $\mathscr{P} \oplus \mathscr{U} \models_{\mathsf{SE}} \mathscr{U}$.

$(\mathsf{P2})_{\mathsf{SE}}$ Suppose that $\mathscr{P} \models_{\mathsf{SE}} \mathscr{U}$ and take some $X \in [\![\mathscr{P}]\!]_{\mathsf{SE}} \subseteq [\![\mathscr{U}]\!]_{\mathsf{SE}}$. Since the preorder assignment is faithful, for all $Y \in [\![\mathscr{U}]\!]_{\mathsf{SE}}$ with $Y \neq X$ we have $X <_\omega^X Y$. Consequently, $\min([\![\mathscr{U}]\!]_{\mathsf{SE}}, \leqslant_\omega^X) = \{X\}$ and so

$$[\![\mathscr{P} \oplus \mathscr{U}]\!]_{\mathsf{SE}} = \bigcup_{X \in [\![\mathscr{P}]\!]_{\mathsf{SE}}} \min\left([\![\mathscr{U}]\!]_{\mathsf{SE}}, \leqslant_\omega^X\right) = \bigcup_{X \in [\![\mathscr{P}]\!]_{\mathsf{SE}}} \{X\} = [\![\mathscr{P}]\!]_{\mathsf{SE}}.$$

$(\mathsf{P3})_{\mathsf{SE}}$ Suppose that both $[\![\mathscr{P}]\!]_{\mathsf{SE}} \neq \emptyset$ and $[\![\mathscr{U}]\!]_{\mathsf{SE}} \neq \emptyset$. Then there is some $X_0 \in [\![\mathscr{P}]\!]_{\mathsf{SE}}$ and also some $Y \in \min([\![\mathscr{U}]\!]_{\mathsf{SE}}, \leqslant_\omega^{X_0})$, so we obtain

$$Y \in \min\left([\![\mathscr{U}]\!]_{\mathsf{SE}}, \leqslant_\omega^{X_0}\right) \subseteq \bigcup_{X \in [\![\mathscr{P}]\!]_{\mathsf{SE}}} \min\left([\![\mathscr{U}]\!]_{\mathsf{SE}}, \leqslant_\omega^X\right) = [\![\mathscr{P} \oplus \mathscr{U}]\!]_{\mathsf{SE}}.$$

Hence, $[\![\mathscr{P} \oplus \mathscr{U}]\!]_{\mathsf{SE}} \neq \emptyset$.

$(\mathsf{P4})_{\mathsf{SE}}$ If $\mathscr{P} \equiv_{\mathsf{SE}} \mathscr{Q}$ and $\mathscr{U} \equiv_{\mathsf{SE}} \mathscr{V}$, then

$$\begin{aligned}\llbracket \mathscr{P} \oplus \mathscr{U} \rrbracket_{\mathsf{SE}} &= \bigcup_{X \in \llbracket \mathscr{P} \rrbracket_{\mathsf{SE}}} \min\left(\llbracket \mathscr{U} \rrbracket_{\mathsf{SE}}, \leqslant_\omega^X\right) = \bigcup_{X \in \llbracket \mathscr{Q} \rrbracket_{\mathsf{SE}}} \min\left(\llbracket \mathscr{V} \rrbracket_{\mathsf{SE}}, \leqslant_\omega^X\right) \\ &= \llbracket \mathscr{Q} \oplus \mathscr{V} \rrbracket_{\mathsf{SE}}.\end{aligned}$$

Therefore, $\mathscr{P} \oplus \mathscr{U} \equiv_{\mathsf{SE}} \mathscr{Q} \oplus \mathscr{V}$.

$(\mathsf{P5})_{\mathsf{SE}}$ Suppose that Y is an SE-model of $(\mathscr{P} \oplus \mathscr{U}) \dot\wedge \mathscr{V}$. Then $Y \in \llbracket \mathscr{V} \rrbracket_{\mathsf{SE}}$ and there is some SE-model X of $\mathscr{P}$ such that Y belongs to $\min(\llbracket \mathscr{U} \rrbracket_{\mathsf{SE}}, \leqslant_\omega^X)$. Consequently, Y also belongs to $\min(\llbracket \mathscr{U} \rrbracket_{\mathsf{SE}} \cap \llbracket \mathscr{V} \rrbracket_{\mathsf{SE}}, \leqslant_\omega^X)$, so Y is an SE-model of $\mathscr{P} \oplus (\mathscr{U} \dot\wedge \mathscr{V})$.

$(\mathsf{P6})_{\mathsf{SE}}$ Assume that $\mathscr{P} \oplus \mathscr{U} \models_{\mathsf{SE}} \mathscr{V}$ and $\mathscr{P} \oplus \mathscr{V} \models_{\mathsf{SE}} \mathscr{U}$. We will prove by contradiction that $\mathscr{P} \oplus \mathscr{U} \models_{\mathsf{SE}} \mathscr{P} \oplus \mathscr{V}$. The other half can be proved similarly.
So suppose that Y is an SE-model of $\mathscr{P} \oplus \mathscr{U}$ but not of $\mathscr{P} \oplus \mathscr{V}$. Then there is some SE-model X of $\mathscr{P}$ such that

$$Y \in \min(\llbracket \mathscr{U} \rrbracket_{\mathsf{SE}}, \leqslant_\omega^X). \tag{A15}$$

At the same time, there must be some SE-model Z of $\mathscr{V}$ such that $Z <_\omega^X Y$. Let Z_0 be minimal w.r.t. $\leqslant_\omega^X$ among all such Z. Then by transitivity of $<_\omega^X$ we obtain that $Z_0 \in \min(\llbracket \mathscr{V} \rrbracket_{\mathsf{SE}}, \leqslant_\omega^X)$ and, consequently, Z_0 is an SE-model of $\mathscr{P} \oplus \mathscr{V}$. By the assumption we now obtain that Z_0 is an SE-model of $\mathscr{U}$. But since $Z_0 <_\omega^X Y$, this is in conflict with (A15).

$(\mathsf{P7})_{\mathsf{SE}}$ Suppose that $\mathscr{P}$ is strongly equivalent to $\|X\|$ for some SE-interpretation X and Y is an SE-model of both $\mathscr{P} \oplus \mathscr{U}$ and $\mathscr{P} \oplus \mathscr{V}$. We will show that Y is an SE-model of $\mathscr{P} \oplus (\mathscr{U} \dot\vee \mathscr{V})$. Let $\mathscr{M} = \llbracket \mathscr{U} \rrbracket_{\mathsf{SE}}$ and $\mathscr{N} = \llbracket \mathscr{V} \rrbracket_{\mathsf{SE}}$. It follows that

$$Y \in \min(\mathscr{M}, \leqslant_\omega^X) \cup \min(\mathscr{M}, \leqslant_\omega^{X^*}) \text{ and } Y \in \min(\mathscr{N}, \leqslant_\omega^X) \cup \min(\mathscr{N}, \leqslant_\omega^{X^*}),$$

so since ω is organised, $Y \in \min(\mathscr{M} \cup \mathscr{N}, \leqslant_\omega^X) \cup \min(\mathscr{M} \cup \mathscr{N}, \leqslant_\omega^{X^*})$. Consequently, Y is an SE-model of $\mathscr{P} \oplus (\mathscr{U} \dot\vee \mathscr{V})$.

$(\mathsf{P8})_{\mathsf{SE}}$ The following sequence of equations establishes the property:

$$\begin{aligned}\llbracket (\mathscr{P} \dot\vee \mathscr{Q}) \oplus \mathscr{U} \rrbracket_{\mathsf{SE}} &= \bigcup_{X \in \llbracket \mathscr{P} \dot\vee \mathscr{Q} \rrbracket_{\mathsf{SE}}} \min\left(\llbracket \mathscr{U} \rrbracket_{\mathsf{SE}}, \leqslant_\omega^X\right) \\ &= \bigcup_{X \in \llbracket \mathscr{P} \rrbracket_{\mathsf{SE}}} \min\left(\llbracket \mathscr{U} \rrbracket_{\mathsf{SE}}, \leqslant_\omega^X\right) \cup \bigcup_{X \in \llbracket \mathscr{Q} \rrbracket_{\mathsf{SE}}} \min\left(\llbracket \mathscr{U} \rrbracket_{\mathsf{SE}}, \leqslant_\omega^X\right) \\ &= \llbracket \mathscr{P} \oplus \mathscr{U} \rrbracket_{\mathsf{SE}} \cup \llbracket \mathscr{Q} \oplus \mathscr{U} \rrbracket_{\mathsf{SE}} \\ &= \llbracket (\mathscr{P} \oplus \mathscr{U}) \dot\vee (\mathscr{Q} \oplus \mathscr{U}) \rrbracket_{\mathsf{SE}} \quad \square\end{aligned}$$

Theorem 21.
Let $\oplus$ be a rule update operator. The following conditions are equivalent:

(a) The operator $\oplus$ satisfies conditions $(\mathsf{P1})_{\mathsf{SE}}$–$(\mathsf{P8})_{\mathsf{SE}}$.
(b) The operator $\oplus$ is characterised by a semi-faithful and organised preorder assignment.
(c) The operator $\oplus$ is characterised by a faithful and organised partial order assignment.

Proof of Theorem 21
Follows from Propositions 38, 40 and 41. $\square$

Appendix B Proofs: properties of the assignment $\mathtt{W}$

Proposition 42
The assignment $\mathtt{W}$ is a preorder assignment.

Proof
Recall that the assignment $\mathtt{W}$ is defined for all $\mathtt{SE}$-interpretations $X = (I, J)$, $Y = (K_1, L_1)$, $Z = (K_2, L_2)$ as follows: $Y \leqslant_{\mathtt{W}}^{X} Z$ if and only if

1. $(L_1 \div J) \subseteq (L_2 \div J)$;
2. If $(L_1 \div J) = (L_2 \div J)$, then $(K_1 \div I) \setminus \Delta \subseteq (K_2 \div I) \setminus \Delta$ where $\Delta = L_1 \div J$.

In order to show that $\mathtt{W}$ is a preorder assignment, we need to prove that given an arbitrary $\mathtt{SE}$-interpretation $X = (I, J)$, $\leqslant_{\mathtt{W}}^{X}$ is a preorder over $\mathcal{X}$. This holds if and only if $\leqslant_{\mathtt{W}}^{X}$ is reflexive and transitive. First we show reflexivity. Take some $\mathtt{SE}$-interpretation $Y = (K, L)$. By definition, $Y \leqslant_{\mathtt{W}}^{X} Y$ holds if and only if

1. $(L \div J) \subseteq (L \div J)$;
2. If $(L \div J) = (L \div J)$, then $(K \div I) \setminus \Delta \subseteq (K \div I) \setminus \Delta$ where $\Delta = L \div J$.

It is not difficult to check that both conditions hold.

To show transitivity, take some $\mathtt{SE}$-interpretations $Y_1 = (K_1, L_1)$, $Y_2 = (K_2, L_2)$, $Y_3 = (K_3, L_3)$ such that $Y_1 \leqslant_{\mathtt{W}}^{X} Y_2$ and $Y_2 \leqslant_{\mathtt{W}}^{X} Y_3$. We need to show that $Y_1 \leqslant_{\mathtt{W}}^{X} Y_3$. According to the definition of $\leqslant_{\mathtt{W}}^{X}$ we obtain

1. $(L_1 \div J) \subseteq (L_2 \div J)$;
2. If $(L_1 \div J) = (L_2 \div J)$, then $(K_1 \div I) \setminus \Delta \subseteq (K_2 \div I) \setminus \Delta$ where $\Delta = L_1 \div J$;

and also

1' $(L_2 \div J) \subseteq (L_3 \div J)$;
2' If $(L_2 \div J) = (L_3 \div J)$, then $(K_2 \div I) \setminus \Delta \subseteq (K_3 \div I) \setminus \Delta$ where $\Delta = L_2 \div J$.

We need to show the following two conditions:

1* $(L_1 \div J) \subseteq (L_3 \div J)$;
2* If $(L_1 \div J) = (L_3 \div J)$, then $(K_1 \div I) \setminus \Delta \subseteq (K_3 \div I) \setminus \Delta$ where $\Delta = L_1 \div J$.

It can be seen that 1* follows from 1. and 1' by transitivity of the subset relation. To show that 2* holds as well, suppose that $(L_1 \div J) = (L_3 \div J)$. Then by 1. and 1' we obtain that $(L_1 \div J) = (L_2 \div J) = (L_3 \div J) = \Delta$ and so by 2. and 2' it holds that

$$(K_1 \div I) \setminus \Delta \subseteq (K_2 \div I) \setminus \Delta \subseteq (K_3 \div I) \setminus \Delta.$$

Consequently, 2* is also satisfied and the proof is finished. □

Lemma 43
Let $X = (I, J)$, $Y = (K_1, L_1)$, $Z = (K_2, L_2)$ be $\mathtt{SE}$-interpretations. Then $Y <_{\mathtt{W}}^{X} Z$ holds if and only if one of the following conditions is satisfied:

(a) $(L_1 \div J) \subsetneq (L_2 \div J)$, or
(b) $(L_1 \div J) = (L_2 \div J)$ and $(K_1 \div I) \setminus \Delta \subsetneq (K_2 \div I) \setminus \Delta$ where $\Delta = L_1 \div J$.

Proof
By definition, $Y <^X_{\mathsf{W}} Z$ holds if and only if $Y \leqslant^X_{\mathsf{W}} Z$ and it is not the case that $Z \leqslant^X_{\mathsf{W}} Y$. This in turn holds if and only if the following two conditions hold

1. $(L_1 \div J) \subseteq (L_2 \div J)$;
2. If $(L_1 \div J) = (L_2 \div J)$, then $(K_1 \div I) \setminus \Delta \subseteq (K_2 \div I) \setminus \Delta$ where $\Delta = L_1 \div J$.

and one of the following conditions also holds:

(i) $(L_2 \div J) \nsubseteq (L_1 \div J)$, or
(ii) $(L_2 \div J) = (L_1 \div J)$ and $(K_2 \div I) \setminus \Delta \nsubseteq (K_1 \div I) \setminus \Delta$ where $\Delta = L_2 \div J$.

It is not difficult to verify that conditions 1., 2. and (i) are together equivalent to (a) and that conditions 1., 2. and (ii) are together equivalent to (b). This concludes our proof. $\square$

Proposition 44
The assignment W is well-defined.

Proof
By definition we need to show that there is a rule update operator $\oplus$ such that for all programs $\mathcal{P}$, $\mathcal{U}$,

$$[\![\mathcal{P} \oplus \mathcal{U}]\!]_{\mathsf{SE}} = \bigcup_{X \in [\![\mathcal{P}]\!]_{\mathsf{SE}}} \min\left([\![\mathcal{U}]\!]_{\mathsf{SE}}, \leqslant^X_{\mathsf{W}}\right).$$

This holds if and only if for every well-defined set of SE-interpretations $\mathcal{M}$ and every SE-interpretation X, the set of SE-interpretations

$$\min\left(\mathcal{M}, \leqslant^X_{\mathsf{W}}\right) \cup \min\left(\mathcal{M}, \leqslant^{X^*}_{\mathsf{W}}\right) \tag{B1}$$

is well-defined. Suppose that Y belongs to (B1). We need to demonstrate that Y^* also belongs to (B1). We consider two cases:

(a) Suppose that $Y \in \min(\mathcal{M}, \leqslant^X_{\mathsf{W}})$. If Y^* belongs to $\min(\mathcal{M}, \leqslant^{X^*}_{\mathsf{W}})$, then we are finished. On the other hand, if Y^* does not belong to $\min(\mathcal{M}, \leqslant^{X^*}_{\mathsf{W}})$, then there must be some $Z \in \mathcal{M}$ such that $Z <^{X^*}_{\mathsf{W}} Y^*$. Let $Y = (K_1, L_1), Z = (K_2, L_2)$ and $X = (I, J)$. By Lemma 43 we know that $Z <^{X^*}_{\mathsf{W}} Y^*$ holds if and only if one of the following conditions is satisfied:

   (a) $(L_2 \div J) \subsetneq (L_1 \div J)$, or
   (b) $(L_2 \div J) = (L_1 \div J)$ and $(K_2 \div J) \setminus \Delta \subsetneq (L_1 \div J) \setminus \Delta$ where $\Delta = L_2 \div J$.

   If (a) is satisfied, then Lemma 43 implies that $Z <^X_{\mathsf{W}} Y$ which is in conflict with the assumption that $Y \in \min(\mathcal{M}, \leqslant^X_{\mathsf{W}})$. So (b) must hold. But in that case we infer that $(K_2 \div J) \setminus \Delta$ is a proper subset of

   $$(L_1 \div J) \setminus \Delta = (L_1 \div J) \setminus (L_1 \div J) = \emptyset,$$

   which is impossible.

(b) Suppose that $Y \in \min(\mathcal{M}, \leqslant^{X^*}_{\mathsf{W}})$ and let $X = (I, J)$, $Y = (K, L)$. First we show that $Y^* \leqslant^{X^*}_{\mathsf{W}} Y$ holds – for this, the following conditions need to be satisfied:

   1. $(L \div J) \subseteq (L \div J)$;

2. If $(L \div J) = (L \div J)$, then $(L \div J) \setminus \Delta \subseteq (K \div J) \setminus \Delta$ where $\Delta = L \div J$.

It is not difficult to verify that both conditions hold.
Thus, since $Y^* \leqslant_{\mathsf{w}}^{X^*} Y$, there can be no $Z \in \mathscr{M}$ with $Z <_{\mathsf{w}}^{X^*} Y^*$ because by transitivity we would obtain $Z <_{\mathsf{w}}^{X^*} Y$ which would be in conflict with the assumption that $Y \in \min(\mathscr{M}, \leqslant_{\mathsf{w}}^{X^*})$. So $Y^* \in \min(\mathscr{M}, \leqslant_{\mathsf{w}}^{X^*})$ and our proof is finished. $\square$

Proposition 45
The assignment W is faithful.

Proof
Take some SE-interpretations $X = (I, J)$, $Y = (K, L)$ such that $Y \neq X$. We need to show that $X <_{\mathsf{w}}^{X} Y$. By Lemma 43 this holds if and only if one of the following conditions is satisfied:

(a) $(J \div J) \subsetneq (L \div J)$, or
(b) $(J \div J) = (L \div J)$ and $(I \div I) \setminus \Delta \subsetneq (K \div I) \setminus \Delta$ where $\Delta = J \div J$.

We consider two cases:

(i) If $L \div J = \emptyset$, then $L = J$ and since $Y \neq X$, we conclude that $K \neq I$. Consequently, the second condition is satisfied because $I \div I = J \div J = \emptyset$ and $K \div I$ is non-empty.
(ii) If $L \div J \neq \emptyset$, then (a) holds since $J \div J = \emptyset$. $\square$

Proposition 46
The assignment W is organised.

Proof
Recall that by definition W is organised if for all SE-interpretations X, Y and all well-defined sets of SE-interpretations $\mathscr{M}, \mathscr{N}$ the following condition is satisfied:

If $Y \in \min(\mathscr{M}, \leqslant_{\mathsf{w}}^{X}) \cup \min(\mathscr{M}, \leqslant_{\mathsf{w}}^{X^*})$ and $Y \in \min(\mathscr{N}, \leqslant_{\mathsf{w}}^{X}) \cup \min(\mathscr{N}, \leqslant_{\mathsf{w}}^{X^*})$,
then $Y \in \min(\mathscr{M} \cup \mathscr{N}, \leqslant_{\mathsf{w}}^{X}) \cup \min(\mathscr{M} \cup \mathscr{N}, \leqslant_{\mathsf{w}}^{X^*})$.

Suppose that $Y \notin \min(\mathscr{M} \cup \mathscr{N}, \leqslant_{\mathsf{w}}^{X}) \cup \min(\mathscr{M} \cup \mathscr{N}, \leqslant_{\mathsf{w}}^{X^*})$. We need to show that at least one of the following holds:

(i) $Y \notin \min(\mathscr{M}, \leqslant_{\mathsf{w}}^{X}) \cup \min(\mathscr{M}, \leqslant_{\mathsf{w}}^{X^*})$;
(ii) $Y \notin \min(\mathscr{N}, \leqslant_{\mathsf{w}}^{X}) \cup \min(\mathscr{N}, \leqslant_{\mathsf{w}}^{X^*})$.

If $Y \notin \mathscr{M}$, then (i) is trivially satisfied. Similarly, if $Y \notin \mathscr{N}$, then (ii) is trivially satisfied. So we can assume that $Y \in \mathscr{M} \cap \mathscr{N}$. It follows from the assumption that there must be some $Z_1, Z_2 \in \mathscr{M} \cup \mathscr{N}$ such that $Z_1 <_{\mathsf{w}}^{X} Y$ and $Z_2 <_{\mathsf{w}}^{X^*} Y$. If Z_1 and Z_2 both belong to $\mathscr{M}$, then (i) is satisfied; if they both belong to $\mathscr{N}$, then (ii) is satisfied. So let's assume, without loss of generality, that $Z_1 \in \mathscr{M}$ and $Z_2 \in \mathscr{N}$. Furthermore, let $X = (I, J)$, $Y = (K, L)$, $Z_1 = (K_1, L_1)$ and $Z_2 = (K_2, L_2)$. It follows from $Z_2 <_{\mathsf{w}}^{X^*} Y$ and Lemma 43 that we need to consider two cases:

(a) If $(L_2 \div J) \subsetneq (L \div J)$, then by Lemma 43 we also have $Z_2 <_{\mathsf{w}}^{X} Y$ and, consequently, (ii) is satisfied.

(b) If $(L_2 \div J) = (L \div J)$ and $(K_2 \div J) \setminus \Delta \subsetneq (K \div J) \setminus \Delta$ where $\Delta = L_2 \div J$, then it follows that $(K \div J) \setminus \Delta \neq \emptyset$ and by using $\Delta = L_2 \div J = L \div J$ we obtain

$$(K \div J) \setminus (L \div J) \neq \emptyset. \tag{B2}$$

Furthermore, from $Z_1 <_{\mathsf{W}}^{X} Y$ we know that one of the following cases occurs:

(a') $(L_1 \div J) \subsetneq (L \div J)$, or

(b') $(L_1 \div J) = (L \div J)$ and $(K_1 \div I) \setminus \Delta \subsetneq (K \div I) \setminus \Delta$, where $\Delta = L_1 \div J$.

We will show that $Z_1^* <_{\mathsf{W}}^{X^*} Y$. By Lemma 43 this holds if and only if one of the following conditions is satisfied:

(a*) $(L_1 \div J) \subsetneq (L \div J)$, or

(b*) $(L_1 \div J) = (L \div J)$ and $(L_1 \div J) \setminus \Delta \subsetneq (K \div J) \setminus \Delta$, where $\Delta = L_1 \div J$.

We see that (a') implies (a*) and (b') together with (B2) implies (b*). Also, since $\mathscr{M}$ is well-defined, we have $Z_1^* \in \mathscr{M}$, so (i) is satisfied. $\square$

Proposition 23.
The assignment W is a well-defined, faithful and organised preorder assignment.

Proof of Proposition 23
Follows by Propositions 42, 44, 45 and 46. $\square$

Appendix C Proofs: Computational complexity of operators characterised by W

Definition 47 (*Truth value assigned by* SE*-interpretation*)
Let X be an SE-interpretation and p an atom. We define the *truth value assigned by X to p* as follows:

$$X(p) = \begin{cases} \mathsf{T} & \text{if } p \in I; \\ \mathsf{U} & \text{if } p \in J \setminus I; \\ \mathsf{F} & \text{if } p \in \mathcal{A} \setminus J. \end{cases}$$

Definition 48 (*Set of relevant atoms*)
Let ϕ be a propositional formula. We inductively define the *set of atoms relevant to* ϕ, denoted by $\mathsf{at}(\phi)$, as follows:

- If ϕ is $\top$ or $\bot$, then $\mathsf{at}(\phi) = \emptyset$;
- If ϕ is an atom p, then $\mathsf{at}(\phi) = \{ p \}$;
- If ϕ is of the form $\neg\psi$, then $\mathsf{at}(\phi) = \mathsf{at}(\psi)$;
- If ϕ is of the form $\psi_1 \wedge \psi_2$, $\psi_1 \vee \psi_2$, $\psi_1 \supset \psi_2$ or $\psi_1 \equiv \psi_2$, then $\mathsf{at}(\phi) = \mathsf{at}(\psi_1) \cup \mathsf{at}(\psi_2)$.

For a logic program $\mathscr{P}$, $\mathsf{at}(\mathscr{P}) = \mathsf{at}(\kappa(\mathscr{P}))$.

Lemma 49
Let $\mathscr{P}$, $\mathscr{U}$ be programs and $\oplus$ a rule update operator characterised by W. If Z belongs to $\min([\![\mathscr{U}]\!]_{\mathrm{SE}}, \leqslant_{\mathsf{W}}^{X})$ for some $X \in [\![\mathscr{P}]\!]_{\mathrm{SE}}$, then $X(p) = Z(p)$ for all $p \in \mathcal{A} \setminus \mathsf{at}(\mathscr{U})$.

Proof
We prove by contradiction. Suppose that our assumptions are satisfied and $X(p) \neq Z(p)$ for some $p \in \mathcal{A} \setminus \mathsf{at}(\mathcal{U})$. Let the SE-interpretation Y be defined as follows:

$$Y(q) = \begin{cases} X(q) & q = p; \\ Z(q) & q \neq p. \end{cases}$$

First note that since Z is an SE-model of $\mathcal{U}$ and Y differs from Z only in the truth value assigned to p, where $p \notin \mathsf{at}(\mathcal{U})$, it follows that Y is also an SE-model of $\mathcal{U}$.

Put $X = (I, J)$, $Y = (K_1, L_1)$ and $Z = (K_2, L_2)$. By assumption, $X(p) \neq Z(p)$, so, by the definition of Y, $Y(p) \neq Z(p)$. Thus, one of the following cases occurs:

(a) If $L_1 \div L_2 = \{ p \}$, then we immediately obtain that $(L_1 \div J) \div (L_2 \div J) = \{ p \}$. Since $Y(p) = X(p)$, we conclude that $p \notin L_1 \div J$ and it follows that

$$(L_1 \div J) \setminus (L_2 \div J) = \emptyset \qquad \text{and} \qquad (L_2 \div J) \setminus (L_1 \div J) = \{ p \} .$$

Consequently, $L_1 \div J \subsetneq L_2 \div J$, so $Y <^X_{\mathsf{w}} Z$, contrary to the assumption that Z belongs to $\min([\![\mathcal{U}]\!]_{\mathsf{SE}}, \leqslant^X_{\mathsf{w}})$.

(b) If $K_1 \div K_2 = \{ p \}$, then we obtain that $(K_1 \div I) \div (K_2 \div I) = \{ p \}$. Since $Y(p) = X(p)$, we conclude that $p \notin K_1 \div I$ and it follows that

$$(K_1 \div I) \setminus (K_2 \div I) = \emptyset \qquad \text{and} \qquad (K_2 \div I) \setminus (K_1 \div I) = \{ p \} .$$

Furthermore, assuming that the previous case does not occur, it follows that $L_1 = L_2$, so for $\Delta = L_1 \div J = L_2 \div J$ it holds that $p \notin \Delta$ because $X(p) = Z(p)$. Consequently, $(K_1 \div I) \setminus \Delta \subsetneq (K_2 \div I) \setminus \Delta$, so $Y <^X_{\mathsf{w}} Z$, contrary to the assumption that Z belongs to $\min([\![\mathcal{U}]\!]_{\mathsf{SE}}, \leqslant^X_{\mathsf{w}})$. $\square$

Definition 50 (*Truth value substitution*)
Let $X = (I, J)$ be an SE-interpretation and p an atom. We define the SE-interpretations $X^{[p:=\mathsf{T}]}$, $X^{[p:=\mathsf{U}]}$ and $X^{[p:=\mathsf{F}]}$ as follows:

$$\begin{aligned} X^{[p:=\mathsf{T}]} &= (I \cup \{ p \}, J \cup \{ p \}), \\ X^{[p:=\mathsf{U}]} &= (I \setminus \{ p \}, J \cup \{ p \}), \\ X^{[p:=\mathsf{F}]} &= (I \setminus \{ p \}, J \setminus \{ p \}). \end{aligned}$$

Lemma 51
Let X, Y, Z be SE-interpretations, p an atom such that $X(p) = Z(p)$ and V a truth value. Then,

$$Y <^X_{\mathsf{w}} Z \qquad \text{implies} \qquad Y^{[p:=\mathsf{V}]} <^{X^{[p:=\mathsf{V}]}}_{\mathsf{w}} Z^{[p:=\mathsf{V}]}.$$

Proof
Put $X = (I, J)$, $Y = (K_1, L_1)$ and $Z = (K_2, L_2)$. The assumption that $X(p) = Z(p)$ implies that

$$p \notin L_2 \div J \qquad \text{and} \qquad p \notin K_2 \div I. \tag{C1}$$

Furthermore, if $Y <^X_{\mathsf{w}} Z$, then, by Lemma 43, one of the following two cases occurs:

(a) If $(L_1 \div J) \subsetneq (L_2 \div J)$, then it follows from (C1) that $p \notin L_1 \div J$ and we obtain the following:

$$(L_1 \cup \{ p \}) \div (J \cup \{ p \}) = L_1 \div J \subsetneq L_2 \div J = (L_2 \cup \{ p \}) \div (J \cup \{ p \}), \tag{C2}$$

$$(L_1 \setminus \{ p \}) \div (J \setminus \{ p \}) = L_1 \div J \subsetneq L_2 \div J = (L_2 \setminus \{ p \}) \div (J \setminus \{ p \}). \tag{C3}$$

Finally, we need to consider two cases depending on V:

(i) If $\mathsf{V} = \mathsf{T}$ or $\mathsf{V} = \mathsf{U}$, then the second components of the SE-interpretations $X^{[p:=\mathsf{V}]}$, $Y^{[p:=\mathsf{V}]}$ and $Z^{[p:=\mathsf{V}]}$ are $J \cup \{ p \}$, $L_1 \cup \{ p \}$ and $L_2 \cup \{ p \}$, respectively. Hence, the desired conclusion follows from (C2) by Lemma 43.

(ii) If $\mathsf{V} = \mathsf{F}$, then the second components of the SE-interpretations $X^{[p:=\mathsf{V}]}$, $Y^{[p:=\mathsf{V}]}$ and $Z^{[p:=\mathsf{V}]}$ are $J \setminus \{ p \}$, $L_1 \setminus \{ p \}$ and $L_2 \setminus \{ p \}$, respectively. Hence, the desired conclusion follows from (C3) by Lemma 43.

(b) If $(L_1 \div J) = (L_2 \div J)$ and $(K_1 \div I) \setminus \Delta \subsetneq (K_2 \div I) \setminus \Delta$ where $\Delta = L_1 \div J$, then $L_1 = L_2$ and it follows from (C1) that $p \notin \Delta$ as well as $p \notin K_1 \div I$, so we obtain the following:

$$(L_1 \cup \{ p \}) \div (J \cup \{ p \}) = (L_2 \cup \{ p \}) \div (J \cup \{ p \}) = \Delta, \tag{C4}$$

$$(L_1 \setminus \{ p \}) \div (J \setminus \{ p \}) = (L_2 \setminus \{ p \}) \div (J \setminus \{ p \}) = \Delta, \tag{C5}$$

$$\begin{aligned}[(K_1 \cup \{ p \}) \div (I \cup \{ p \})] \setminus \Delta &= (K_1 \div I) \setminus \Delta \\ &\subsetneq (K_2 \div I) \setminus \Delta = [(K_2 \cup \{ p \}) \div (I \cup \{ p \})] \setminus \Delta,\end{aligned} \tag{C6}$$

$$\begin{aligned}[(K_1 \setminus \{ p \}) \div (I \setminus \{ p \})] \setminus \Delta &= (K_1 \div I) \setminus \Delta \\ &\subsetneq (K_2 \div I) \setminus \Delta = [(K_2 \setminus \{ p \}) \div (I \setminus \{ p \})] \setminus \Delta.\end{aligned} \tag{C7}$$

Finally, we need to use Lemma 43, considering three cases depending on V:

(i) If $\mathsf{V} = \mathsf{T}$, then the desired conclusion follows from (C4) and (C6).

(ii) If $\mathsf{V} = \mathsf{U}$, then the desired conclusion follows from (C4) and (C7).

(iii) If $\mathsf{V} = \mathsf{F}$, then the desired conclusion follows from (C5) and (C7). □

Lemma 52

Let $\mathscr{P}$, $\mathscr{U}$ be programs, p an atom with $p \notin \mathsf{at}(\mathscr{P}) \cup \mathsf{at}(\mathscr{U})$, $\oplus$ a rule update operator characterised by W and Z, Z' be SE-interpretations such that $Z = Z'^{[p:=\mathsf{V}]}$ for some truth value V. Then,

$$Z \in [\![\mathscr{P} \oplus \mathscr{U}]\!]_{\mathsf{SE}} \qquad \text{if and only if} \qquad Z' \in [\![\mathscr{P} \oplus \mathscr{U}]\!]_{\mathsf{SE}}.$$

Proof

We prove the direct implication, the converse one follows by the symmetry of the claim.

Suppose that $Z \in [\![\mathscr{P} \oplus \mathscr{U}]\!]_{\mathsf{SE}}$ but $Z' \notin [\![\mathscr{P} \oplus \mathscr{U}]\!]_{\mathsf{SE}}$. Then there is some SE-interpretation $X \in [\![\mathscr{P}]\!]_{\mathsf{SE}}$ such that Z belongs to $\min([\![\mathscr{U}]\!]_{\mathsf{SE}}, \leqslant^X_{\mathsf{W}})$. It follows from Lemma 49 that

$$X(p) = Z(p) = \mathsf{V}.$$

Put $Z'(p) = \mathsf{V}'$ and let $X' = X^{[p:=\mathsf{V}']}$. Since X' differs from X only in the truth value assigned to p and $p \notin \mathsf{at}(\mathscr{P})$, it follows that $X' \in [\![\mathscr{P}]\!]_{\mathsf{SE}}$. Thus, there exists some

SE-interpretation Y' such that $Y' <_{\mathsf{W}}^{X'} Z'$ and by Lemma 51 we conclude that

$$Y'^{[p:=\mathsf{V}]} <_{\mathsf{W}}^{X'^{[p:=\mathsf{V}]}} Z'^{[p:=\mathsf{V}]}.$$

It remains to observe that $X'^{[p:=\mathsf{V}]} = X$ and $Z'^{[p:=\mathsf{V}]} = Z$, so for $Y = Y'^{[p:=\mathsf{V}]}$ we have

$$Y <_{\mathsf{W}}^{X} Z.$$

Since Y differs from Y' only in the truth value assigned to p and $p \notin \mathsf{at}(\mathscr{U})$, it follows that $Y \in [\![\mathscr{U}]\!]_{\mathsf{SE}}$ – a conflict with the assumption that Z belongs to $\min([\![\mathscr{U}]\!]_{\mathsf{SE}}, \leqslant_{\mathsf{W}}^{X})$. $\square$

Corollary 53
Let $\mathscr{P}$, $\mathscr{U}$ be programs, $\oplus$ a rule update operator characterised by W and Z, Z' be SE-interpretations such that $Z(p) = Z'(p)$ for all $p \in \mathsf{at}(\mathscr{P}) \cup \mathsf{at}(\mathscr{U})$. Then,

$$Z \in [\![\mathscr{P} \oplus \mathscr{U}]\!]_{\mathsf{SE}} \qquad \text{if and only if} \qquad Z' \in [\![\mathscr{P} \oplus \mathscr{U}]\!]_{\mathsf{SE}}.$$

Proof
Suppose that

$$\mathcal{A} \setminus (\mathsf{at}(\mathscr{P}) \cup \mathsf{at}(\mathscr{U})) = \{ p_1, p_2, \ldots, p_n \}$$

and construct a sequence of SE-interpretations $Z_0, Z_1, \ldots, Z_n$ as follows: $Z_0 = Z$ and $Z_{i+1} = Z_i^{[p_i:=Z'(p_i)]}$ for all i with $0 \leqslant i < n$. Clearly, $Z_n = Z'$ and Lemma 52 can be used n times, for each pair (Z_i, Z_{i+1}), to infer the desired result. $\square$

Lemma 54
Let $\mathscr{P}$ be a set of facts, $\mathscr{U}$ a program such that $\mathsf{at}(\mathscr{U}) \subseteq \mathsf{at}(\mathscr{P})$, $\oplus$ a rule update operator characterised by W and Z an SE-interpretation from $[\![\mathscr{P} \oplus \mathscr{U}]\!]_{\mathsf{SE}}$. Then for every atom p with $(p.) \in \mathscr{P}$ it holds that $Z(p) \neq \mathsf{U}$.

Proof
Suppose that Z belongs to $[\![\mathscr{P} \oplus \mathscr{U}]\!]_{\mathsf{SE}}$, put $Z = (K, L)$ and let

$$Y = (K \cap \mathsf{at}(\mathscr{P}), L \cap \mathsf{at}(\mathscr{P})).$$

It follows by Corollary 53 that Y belongs to $[\![\mathscr{P} \oplus \mathscr{U}]\!]_{\mathsf{SE}}$. Thus, there exists some SE-interpretation $X \in [\![\mathscr{P}]\!]_{\mathsf{SE}}$ such that Y belongs to $\min([\![\mathscr{U}]\!]_{\mathsf{SE}}, \leqslant_{\mathsf{W}}^{X})$. Also, using Lemma 49 we conclude that X assigns truth values as follows:

$$X(q) = \begin{cases} \mathsf{T} & (q.) \in \mathscr{P}; \\ \mathsf{F} & (\sim q.) \in \mathscr{P}; \\ \mathsf{F} & q \in \mathcal{A} \setminus \mathsf{at}(\mathscr{P}). \end{cases}$$

In other words, X is of the form (J, J) where $J = \{ q \in \mathcal{A} \mid (q.) \in \mathscr{P} \}$. Furthermore, since Y belongs to $[\![\mathscr{U}]\!]_{\mathsf{SE}}$, $Y^* = (L \cap \mathsf{at}(\mathscr{P}), L \cap \mathsf{at}(\mathscr{P}))$ also belongs there.

We proceed by contradiction: Suppose that $Z(p) = \mathsf{U}$ for some atom p with $(p.) \in \mathscr{P}$. Then $p \in L \setminus K$, $p \in \mathsf{at}(\mathscr{P})$ and $p \in J$ and we reach a conflict because $Y^* <_{\mathsf{W}}^{X} Y$ follows by Lemma 43 from the fact that

$$\begin{aligned} &[(L \cap \mathsf{at}(\mathscr{P})) \div J] \setminus [(L \cap \mathsf{at}(\mathscr{P})) \div J] = \emptyset \\ &\quad \subsetneq \{ p \} \subseteq [(K \cap \mathsf{at}(\mathscr{P})) \div J] \setminus [(L \cap \mathsf{at}(\mathscr{P})) \div J]. \end{aligned} \quad \square$$

Lemma 55
Let $\mathscr{P}$, $\mathscr{U}$ be programs, $\oplus$ a rule update operator characterised by W, $\diamond$ a belief update operator characterised by W and L an interpretation. Then,

$$(L, L) \in [\![\mathscr{P} \oplus \mathscr{U}]\!]_{\mathsf{SE}} \qquad \text{if and only if} \qquad L \in [\![\kappa(\mathscr{P}) \diamond \kappa(\mathscr{U})]\!].$$

Proof
Suppose that $(L, L) \in [\![\mathscr{P} \oplus \mathscr{U}]\!]_{\mathsf{SE}}$. Then (L, L) belongs to $\min([\![\mathscr{U}]\!]_{\mathsf{SE}}, \leqslant^X_{\mathsf{W}})$ for some $X = (I, J) \in [\![\mathscr{P}]\!]_{\mathsf{SE}}$. Since $[\![\mathscr{P}]\!]_{\mathsf{SE}}$ is a well-defined set of SE-interpretations, we conclude that $(J, J) \in [\![\mathscr{P}]\!]_{\mathsf{SE}}$ and, consequently, $J \models \mathscr{P}$. We will prove that $L \in \min([\![\kappa(\mathscr{U})]\!], \leqslant^J_{\mathsf{W}})$. Suppose that this is not the case, i.e. there is some $L' \in [\![\kappa(\mathscr{U})]\!]$ such that $L' <^J_{\mathsf{W}} L$. In other words, $L' \div J \subsetneq L \div J$. It follows that (L', L') is an SE-model of $\mathscr{U}$ and by Lemma 43 we conclude that $(L', L') <^X_{\mathsf{W}} (L, L)$, contrary to the assumption that (L, L) belongs to $\min([\![\mathscr{U}]\!]_{\mathsf{SE}}, \leqslant^X_{\mathsf{W}})$.

To prove the converse implication, assume that $L \in [\![\kappa(\mathscr{P}) \diamond \kappa(\mathscr{U})]\!]$. Then there is some interpretation J with $J \models \mathscr{P}$ such that $L \in \min([\![\kappa(\mathscr{U})]\!], \leqslant^J_{\mathsf{W}})$. It follows that $X = (J, J) \in [\![\mathscr{P}]\!]_{\mathsf{SE}}$ and $Z = (L, L) \in [\![\mathscr{U}]\!]_{\mathsf{SE}}$. Our goal is to prove that $Z \in \min([\![\mathscr{U}]\!]_{\mathsf{SE}}, \leqslant^X_{\mathsf{W}})$. Suppose that this is not the case, i.e. there is some $Z' = (K', L') \in [\![\mathscr{U}]\!]_{\mathsf{SE}}$ such that $Z' <^X_{\mathsf{W}} Z$. Note that since $[\![\mathscr{U}]\!]_{\mathsf{SE}}$ is a well-defined set of SE-interpretations, it follows that $(L', L') \in [\![\mathscr{U}]\!]_{\mathsf{SE}}$ and thus $L' \models \mathscr{U}$. By Lemma 43, one of the following conditions is then satisfied:

(a) If $L' \div J \subsetneq L \div J$, then we obtain $L' <^J_{\mathsf{W}} L$, contrary to the assumption that L belongs to $\min([\![\kappa(\mathscr{U})]\!], \leqslant^J_{\mathsf{W}})$.
(b) The case when $L' \div J = L \div J$ and $(K' \div J) \setminus \Delta \subsetneq (L \div J) \setminus \Delta$, where $\Delta = L \div J$, is impossible because the set $(L \div J) \setminus \Delta$ is empty. □

Proposition 56
Let $\mathscr{P}$ be a set of facts, $\mathscr{Q}$ and $\mathscr{U}$ be programs such that $\mathscr{Q} \subseteq \mathscr{P}$ and $\mathsf{at}(\mathscr{U}) \subseteq \mathsf{at}(\mathscr{P})$, $\oplus$ a rule update operator characterised by W and $\diamond$ a belief update operator characterised by W. Then,

$$\mathscr{P} \oplus \mathscr{U} \models_{\mathsf{SE}} \mathscr{Q} \qquad \text{if and only if} \qquad \kappa(\mathscr{P}) \diamond \kappa(\mathscr{U}) \models \kappa(\mathscr{Q}).$$

Proof
First suppose that $\mathscr{P} \oplus \mathscr{U} \models_{\mathsf{SE}} \mathscr{Q}$ and take some $L \in [\![\kappa(\mathscr{P}) \diamond \kappa(\mathscr{U})]\!]$. We need to prove that $L \models \mathscr{Q}$. It follows from Lemma 55 that $(L, L) \in [\![\mathscr{P} \oplus \mathscr{U}]\!]_{\mathsf{SE}}$ and our assumption implies that $(L, L) \models \mathscr{Q}$. This means that $L \models \mathscr{Q}$, so we reached the desired conclusion.

For the converse implication, suppose that $\kappa(\mathscr{P}) \diamond \kappa(\mathscr{U}) \models \kappa(\mathscr{Q})$ and take some $(K, L) \in [\![\mathscr{P} \oplus \mathscr{U}]\!]_{\mathsf{SE}}$. Our goal is to prove that $(K, L) \models \mathscr{Q}$. Since the set of SE-interpretations $[\![\mathscr{P} \oplus \mathscr{U}]\!]_{\mathsf{SE}}$ is well-defined, we obtain that $(L, L) \in [\![\mathscr{P} \oplus \mathscr{U}]\!]_{\mathsf{SE}}$ and by Lemma 55 it follows that $L \in [\![\kappa(\mathscr{P}) \diamond \kappa(\mathscr{U})]\!]$. By our assumption we infer that $L \models \mathscr{Q}$. Thus, for every positive fact $(p.)$ from $\mathscr{Q}$ it holds that $p \in L$ and due to Lemma 54 also $p \in K$. Therefore, $(K, L) \models (p.)$. Similarly, for every negative fact $(\sim p.)$ from $\mathscr{Q}$ it holds that $p \notin L$ and, hence, $(K, L) \models (\sim p.)$. Consequently, $(K, L) \models \mathscr{Q}$ as desired. □

Theorem 25 (Computational complexity of rule updates characterised by W*).*
Let $\oplus$ be a rule update operator characterised by W. Deciding whether $\mathscr{P} \oplus \mathscr{U} \models_{\mathrm{SE}} \mathscr{Q}$ for programs $\mathscr{P}$, $\mathscr{U}$, $\mathscr{Q}$ is Π_2^{P}-complete. Hardness holds even if $\mathscr{P}$ is a set of positive facts, $\mathscr{U}$ is a non-disjunctive program and $\mathscr{Q}$ contains a single fact from $\mathscr{P}$.

Proof of Theorem 25
Hardness can be shown by reducing the problem of query answering for Winslett's belief update semantics to the problem of query answering for $\oplus$. To do this, we rely on some specifics of the proof of Theorem 6 as it is presented in Eiter and Gottlob (1992). More specifically, Lemma 6.2 (cf. page 250 of Eiter and Gottlob 1992) shows Π_2^{P}-hardness of Winslett's belief update semantics by taking an instance

$$F = \forall x_1, \dots, x_m \exists y_1, \dots, y_n : v$$

of $\mathrm{QBF}_{2,\forall}$ and constructing propositional formulae ϕ, μ and ψ such that

$$F \text{ is valid} \qquad \text{if and only if} \qquad \phi \diamond \mu \models \psi. \tag{C8}$$

In the following we reproduce the definition of ϕ, μ and ψ in order to pinpoint their syntactic structure. Then we show how they can be encoded as logic programs $\mathscr{P}$, $\mathscr{U}$ and $\mathscr{Q}$ such that

$$\phi \diamond \mu \models \psi \qquad \text{if and only if} \qquad \mathscr{P} \oplus \mathscr{U} \models_{\mathrm{SE}} \mathscr{Q}. \tag{C9}$$

However, we omit the proof of the equivalence (C8) and refer the interested reader to Eiter and Gottlob (1992) for further details.

Formulae ϕ, μ and ψ can be defined as follows:

$$\begin{aligned}
\phi &= x_1 \wedge \cdots \wedge x_m \wedge z_1 \wedge \cdots \wedge z_m \wedge y_1 \wedge \cdots \wedge y_n \wedge r, \\
\mu &= (x_1 \equiv \neg z_1) \wedge \cdots \wedge (x_m \equiv \neg z_m) \wedge (r \supset v) \wedge ((y_1 \vee \cdots \vee y_n) \supset r), \\
\psi &= r,
\end{aligned}$$

where $z_1, \dots, z_m$ and r are fresh propositional variables. Moreover, we can assume without loss of generality that v is in conjunctive normal form, i.e.

$$v = \bigwedge_{i=1}^{s} (p_{i,1} \vee \cdots \vee p_{i,t_i} \vee \neg q_{i,1} \vee \cdots \vee \neg q_{i,u_i})$$

where $p_{i,j}$ and $q_{i,k}$ belong to $\{ x_1, \dots, x_m, y_1, \dots, y_n \}$ for all i, j, k. We construct programs $\mathscr{P}$, $\mathscr{U}$ and $\mathscr{Q}$ as follows:

$$\begin{aligned}
\mathscr{P} &= \{ (x_i.) \mid 1 \leqslant i \leqslant m \} \cup \{ (z_i.) \mid 1 \leqslant i \leqslant m \} \cup \{ (y_i.) \mid 1 \leqslant i \leqslant n \} \cup \{ (r.) \}, \\
\mathscr{U} &= \{ (x_i \leftarrow {\sim} z_i.), ({\sim} z_i \leftarrow x_i.) \mid 1 \leqslant i \leqslant m \} \\
&\qquad \cup \{ (\bot \leftarrow {\sim} p_{i,1}, \dots, {\sim} p_{i,t_i}, q_{i,1}, \dots, q_{i,u_i}, r.) \mid 1 \leqslant i \leqslant s \} \\
&\qquad \cup \{ (r \leftarrow y_i.) \mid 1 \leqslant i \leqslant n \}, \\
\mathscr{Q} &= \{ (r.) \}.
\end{aligned}$$

It is not difficult to verify that $\kappa(\mathscr{P}) \equiv \phi$, $\kappa(\mathscr{U}) \equiv \mu$ and $\kappa(\mathscr{Q}) \equiv \psi$, so it follows from postulate (B4) and Proposition 56 that (C9) is satisfied. Together with (C8) this implies that query answering for rule update operators characterised by W is Π_2^{P}-hard.

To verify membership to Π_2^{P}, consider the following non-deterministic polynomial algorithm with an NP oracle, analogous to the one for Winslett's belief update semantics (cf. proof of Theorem 6.4 on page 252 in Eiter and Gottlob 1992): To prove that $\mathscr{P} \oplus \mathscr{U} \not\models_{\mathsf{SE}} \mathscr{Q}$, consider only atoms from $\mathsf{at}(\mathscr{P}) \cup \mathsf{at}(\mathscr{U}) \cup \mathsf{at}(\mathscr{Q})$ (this can be done due to Corollary 53), guess some SE-interpretations X and Y, check in polynomial time that $X \in [\![\mathscr{P}]\!]_{\mathsf{SE}}$, $Y \in [\![\mathscr{U}]\!]_{\mathsf{SE}}$ and $Y \notin [\![\mathscr{Q}]\!]_{\mathsf{SE}}$ and invoke the NP oracle to check that there is no $Z \in [\![\mathscr{U}]\!]_{\mathsf{SE}}$ such that $Z <_{\mathsf{W}}^{X} Y$. □

Lemma 57
Let $\mathscr{U}$ be a definite program. Then for all interpretations I, J it holds that,

$$(I, J) \in [\![\mathscr{U}]\!]_{\mathsf{SE}} \qquad \text{if and only if} \qquad I \subseteq J \wedge I \models \kappa(\mathscr{U}) \wedge J \models \kappa(\mathscr{U}).$$

Proof
Follows from the fact that since $\mathscr{U}$ is definite, $\mathscr{U}^K = \mathscr{U}$ for any interpretation K. □

Theorem 26 (Computational complexity of definite rule updates characterised by W).
Let $\oplus$ be a rule update operator characterised by W. Deciding whether $\mathscr{P} \oplus \mathscr{U} \models_{\mathsf{SE}} \mathscr{Q}$ for definite programs $\mathscr{P}$, $\mathscr{U}$, $\mathscr{Q}$ is co-NP-complete. Hardness holds even if $\mathscr{P}$ is a set of facts and $\mathscr{Q}$ contains a single fact from $\mathscr{P}$.

Proof of Theorem 26
Hardness follows by reducing the co-NP-complete problem of query answering for Horn formulae under Winslett's belief update semantics. More specifically, Theorem 7 shows that deciding whether $\phi \diamond \mu \models \psi$, where $\diamond$ is a belief update operator characterised by W, is co-NP-hard even when ϕ is a conjunction of objective literals, μ is a Horn formula and ψ is one of the literals in ϕ. It is straightforward to construct a set of facts $\mathscr{P}$, a definite program $\mathscr{U}$ and a program $\mathscr{Q}$ containing a single fact from $\mathscr{P}$ such that $\kappa(\mathscr{P}) \equiv \phi$, $\kappa(\mathscr{U}) \equiv \mu$ and $\kappa(\mathscr{Q}) \equiv \psi$. Finally, it follows from postulate (B4) and Proposition 56 that

$$\mathscr{P} \oplus \mathscr{U} \models_{\mathsf{SE}} \mathscr{Q} \qquad \text{if and only if} \qquad \phi \oplus \mu \models \psi,$$

which concludes the proof of co-NP-hardness of query answering for $\oplus$.

To verify membership to co-NP, consider the following non-deterministic polynomial algorithm, analogous to the one for Winslett's belief update semantics for Horn formulae (cf. proof of Theorem 7.2 on page 259 in Eiter and Gottlob 1992): To prove that $\mathscr{P} \oplus \mathscr{U} \not\models_{\mathsf{SE}} \mathscr{Q}$, consider only atoms from $\mathcal{A}' = \mathsf{at}(\mathscr{P}) \cup \mathsf{at}(\mathscr{U}) \cup \mathsf{at}(\mathscr{Q})$ (this can be done due to Corollary 53), guess some SE-interpretations $X = (I, J)$ and $Y = (K, L)$ and check in polynomial time that $X \in [\![\mathscr{P}]\!]_{\mathsf{SE}}$, $Y \in [\![\mathscr{U}]\!]_{\mathsf{SE}}$ and $Y \notin [\![\mathscr{Q}]\!]_{\mathsf{SE}}$. It remains to check that there is no SE-interpretation $Z \in [\![\mathscr{U}]\!]_{\mathsf{SE}}$ such that $Z <_{\mathsf{W}}^{X} Y$. This can be performed in polynomial time by using Lemma 57 as follows. Put $\Delta = L \div J$ and $\Delta' = (K \div I) \setminus \Delta$ and let for every atom p,

$$t(p) = \begin{cases} p & J \models p; \\ \neg p & J \not\models p; \end{cases} \qquad s(p) = \begin{cases} p & I \models p; \\ \neg p & I \not\models p. \end{cases}$$

It follows from Lemma 57 and from the definition of $\leqslant_{\mathsf{w}}^{X}$ that it suffices to verify that for every $p \in \Delta$ and every $q \in \Delta'$, both of the Horn formulae

$$\kappa(\mathscr{U}) \wedge t(p) \wedge \bigwedge_{r \in \mathcal{A}' \setminus \Delta} t(r) \qquad \text{and} \qquad \kappa(\mathscr{U}) \wedge s(q) \wedge \bigwedge_{r \in \mathcal{A}' \setminus \Delta'} s(r)$$

are not satisfiable. $\square$

References

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

Alferes, J. J., Banti, F., Brogi, A. and Leite, J. A. 2005. The refined extension principle for semantics of dynamic logic programming. *Studia Logica 79,* 1, 7–32.

Alferes, J. J., Brogi, A., Leite, J. A. and Pereira, L. M. 2003. An evolvable rule-based e-mail agent. In *Proc. of the 11th Portuguese Conference Artificial Intelligence (EPIA 2003)*, F. Moura-Pires and S. Abreu, Eds. Lecture Notes in Computer Science, vol. 2902. Springer, Beja, Portugal, 394–408.

Alferes, J. J., Leite, J. A., Pereira, L. M., Przymusinska, H. and Przymusinski, T. C. 2000. Dynamic updates of non-monotonic knowledge bases. *The Journal of Logic Programming 45,* 1–3(September/October), 43–70.

Alferes, J. J. and Pereira, L. M. 1996. Update-programs can update programs. In *Non-Monotonic Extensions of Logic Programming (NMELP '96), Selected Papers*, J. Dix, L. M. Pereira and T. C. Przymusinski, Eds. Lecture Notes in Computer Science, vol. 1216. Springer, Bad Honnef, Germany, 110–131.

Apt, K. R., Blair, H. A. and Walker, A. 1988. Towards a theory of declarative knowledge. In *Foundations of Deductive Databases and Logic Programming*, J. Minker, Ed. Morgan Kaufmann, San Francisco, CA, USA. 89–148.

Baral, C. 2003. *Knowledge Representation, Reasoning, and Declarative Problem Solving*. Cambridge University Press, Cambridge, United Kingdom.

Cabalar, P. and Ferraris, P. 2007. Propositional theories are strongly equivalent to logic programs. *Theory and Practice of Logic Programming 7,* 6, 745–759.

Delgrande, J. P. 2010. A program-level approach to revising logic programs under the answer set semantics. *Theory and Practice of Logic Programming, 26th Intl. Conference on Logic Programming (ICLP'10) Special Issue 10,* 4–6 (July), 565–580.

Delgrande, J. P., Schaub, T. and Tompits, H. 2007. A preference-based framework for updating logic programs. In *Proc. of the 9th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2007)*, C. Baral, G. Brewka and J. S. Schlipf, Eds. Lecture Notes in Computer Science, vol. 4483. Springer, Tempe, AZ, USA, 71–83.

Delgrande, J. P., Schaub, T., Tompits, H. and Woltran, S. 2008. Belief revision of logic programs under answer set semantics. In *Proc. of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR 2008)*, G. Brewka and J. Lang, Eds. AAAI Press, Sydney, Australia, 411–421.

Dix, J. 1995a. A classification theory of semantics of normal logic programs: I. Strong properties. *Fundamenta Informaticae 22,* 3, 227–255.

Dix, J. 1995b. A classification theory of semantics of normal logic programs: II. Weak properties. *Fundamenta Informaticae 22,* 3, 257–288.

Eiter, T., Fink, M., Sabbatini, G. and Tompits, H. 2002. On properties of update sequences based on causal rejection. *Theory and Practice of Logic Programming 2,* 6, 721–777.

Eiter, T. and Gottlob, G. 1992. On the complexity of propositional knowledge base revision, updates, and counterfactuals. In *Proc. of the 11th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 1992)*. ACM Press, San Diego, CA, USA, 261–273.

Gelfond, M. and Lifschitz, V. 1988. The stable model semantics for logic programming. In *Proc. of the 5th International Conference and Symposium on Logic Programming (ICLP/SLP 1988)*, R. A. Kowalski and K. A. Bowen, Eds. MIT Press, Seattle, Washington, 1070–1080.

Herzig, A. and Rifi, O. 1999. Propositional belief base update and minimal change. *Artificial Intelligence 115,* 1, 107–138.

Heyting, A. 1930. Die formalen Regeln der intuitionistischen Logik. *Sitzungsberichte der Preussischen Akademie der Wissenschaften*, 42–56. Reprint in *Logik-Texte: Kommentierte Auswahl zur Geschichte der Modernen Logik*, Akademie-Verlag, Berlin, Germany, 1986.

Ilic, M., Leite, J. and Slota, M. 2008. Explicit dynamic user profiles for a collaborative filtering recommender system. In *Proc. of the 11th Ibero-American Conference on Artificial Intelligence (IBERAMIA'08)*, H. Geffner, R. Prada, I. M. Alexandre and N. David, Eds. Vol. LNAI 5290. Springer-Verlag, Berlin, Heidelberg, 352–361.

Katsuno, H. and Mendelzon, A. O. 1991. On the difference between updating a knowledge base and revising it. In *Proc. of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, J. F. Allen, R. Fikes, and E. Sandewall, Eds. Morgan Kaufmann Publishers, Cambridge, MA, USA, 387–394.

Katsuno, H. and Mendelzon, A. O. 1992. Propositional knowledge base revision and minimal change. *Artificial Intelligence 52,* 3, 263–294.

Keller, A. M. and Winslett, M. 1985. On the use of an extended relational model to handle changing incomplete information. *IEEE Transactions on Software Engineering 11,* 7, 620–633.

Krümpelmann, P. and Kern-Isberner, G. 2010. On belief dynamics of dependency relations for extended logic programs. In *Proc. of the 13th International Workshop on Non-Monotonic Reasoning*. Toronto, Canada.

Leite, J. A. 2003. *Evolving Knowledge Bases*: Frontiers of Artificial Intelligence and Applications, xviii + 307 p. Hardcover, Amsterdam, The Netherlands, vol. 81. IOS Press.

Leite, J. A. and Pereira, L. M. 1998. Generalizing updates: From models to programs. In *Proc. of the 3rd International Workshop on Logic Programming and Knowledge Representation (LPKR '97), October 17, 1997, Port Jefferson, New York, USA*, J. Dix, L. M. Pereira and T. C. Przymusinski, Eds. Lecture Notes in Computer Science, vol. 1471. Springer, New York, USA, 224–246.

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

Łukasiewicz, J. 1941. Die Logik und das Grundlagenproblem. In *Les Entretiens de Zürich sue les Fondements et la méthode des sciences mathématiques 1938*. Zürich, 82–100.

Meyer, A. R. and Stockmeyer, L. J. 1972. The equivalence problem for regular expressions with squaring requires exponential space. In *Proc. of the 13th Annual Symposium on Switching and Automata Theory (SWAT)* (October 25–27). IEEE Computer Society, College Park, Maryland, USA, 125–129.

Osorio, M. and Cuevas, V. 2007. Updates in answer set programming: An approach based on basic structural properties. *Theory and Practice of Logic Programming 7,* 4, 451–479.

Pearce, D. 1997. A new logical characterisation of stable models and answer sets. In *Proc. of the 6th Workshop on Non-Monotonic Extensions of Logic Programming (NMELP '96)*, J. Dix, L. M. Pereira and T. C. Przymusinski, Eds. Lecture Notes in Computer Science, vol. 1216. Springer, Bad Honnef, Germany, 57–70.

Saias, J. and Quaresma, P. 2004. A methodology to create legal ontologies in a logic programming based web information retrieval system. *Artificial Intelligence and Law 12,* 4, 397–417.

Sakama, C. and Inoue, K. 2003. An abductive framework for computing knowledge base updates. *Theory and Practice of Logic Programming 3,* 6, 671–713.

Šefránek, J. 2006. Irrelevant updates and nonmonotonic assumptions. In *Proc. of the 10th European Conference on Logics in Artificial Intelligence (JELIA 2006)*, M. Fisher, W. van der Hoek, B. Konev and A. Lisitsa, Eds. Lecture Notes in Computer Science, vol. 4160. Springer, Liverpool, UK, 426–438.

Šefránek, J. 2011. Static and dynamic semantics: Preliminary report. In *Mexican International Conference on Artificial Intelligence*, Puebla, 36–42.

Siska, J. 2006. Dynamic logic programming and world state evaluation in computer games. In *Proc. of the 20th Workshop on Logic Programming*, M. Fink, H. Tompits, and S. Woltran, Eds. INFSYS Research Report, vol. 1843-06-02. Technische Universität Wien, Austria, Vienna, Austria, 64–70.

Slota, M. and Leite, J. 2010. On semantic update operators for answer-set programs. In *Proc. of the 19th European Conference on Artificial Intelligence (ECAI 2010)*, H. Coelho, R. Studer and M. Wooldridge, Eds. Frontiers in Artificial Intelligence and Applications, vol. 215. IOS Press, Lisbon, Portugal, 957–962.

Slota, M. and Leite, J. 2012. Robust equivalence models for semantic updates of answer-set programs. In *Proc. of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR 2012)*, G. Brewka, T. Eiter and S. A. McIlraith, Eds. AAAI Press, Rome, Italy, 158–168.

Slota, M., Leite, J. and Swift, T. 2011. Splitting and updating hybrid knowledge bases. *Theory and Practice of Logic Programming, 27th Intl. Conference on Logic Programming (ICLP'11) Special Issue 11,* 4–5, 801–819.

Stockmeyer, L. J. 1976. The polynomial-time hierarchy. *Theoretical Computer Science 3,* 1, 1–22.

Turner, H. 2003. Strong equivalence made easy: Nested expressions and weight constraints. *Theory and Practice of Logic Programming 3,* 4–5, 609–622.

Winslett, M. 1990. *Updating Logical Databases.* Cambridge University Press, New York, USA.

Zhang, Y. 2006. Logic program-based updates. *ACM Transactions on Computational Logic 7,* 3, 421–472.

Zhang, Y. and Foo, N. Y. 2005. A unified framework for representing logic program updates. In *Proc. of the 20th National Conference on Artificial Intelligence (AAAI 2005)*, M. M. Veloso and S. Kambhampati, Eds. AAAI Press/The MIT Press, Pittsburgh, Pennsylvania, USA, 707–713.