Exception-Based Knowledge Updates[☆]

Martin Slota, João Leite[∗]

NOVA LINCS & Departamento de Informática, Universidade Nova de Lisboa,
Quinta da Torre, 2829-516 Caparica, Portugal

Abstract

Existing methods for dealing with *knowledge updates* differ greatly depending on the underlying knowledge representation formalism. When Classical Logic is used, updates are typically performed by manipulating the knowledge base on the model-theoretic level. On the opposite side of the spectrum stand the semantics for updating Answer-Set Programs that need to rely on rule syntax. Yet, a *unifying perspective* that could embrace both these branches of research is of great importance as it enables a deeper understanding of all involved methods and principles and creates room for their cross-fertilisation, ripening and further development. Furthermore, from a more pragmatic viewpoint, such a unification is a necessary step in addressing *updates of hybrid knowledge bases* consisting of both a classical and a rule component.

This paper bridges the seemingly irreconcilable approaches to updates. It introduces a novel monotonic characterisation of rules, dubbed *RE-models*, and shows it to be a more suitable semantic foundation for rule updates than SE-models. Then it proposes a generic scheme for specifying semantic rule update operators, based on the idea of viewing a program as the *set of sets of RE-models* of its rules; updates are performed by introducing additional interpretations – *exceptions* – to the sets of RE-models of rules in the original program. The introduced scheme is then used to define particular rule update operators that are closely related to both classical update principles and traditional approaches to rules updates, enjoying a range of plausible syntactic as well as semantic properties. In addition, these operators serve as a basis for a solution to the long-standing problem of *state condensing* for two of the foundational rule update semantics, showing how they can be equivalently defined as binary operators on some class of logic programs.

Finally, the essence of these ideas is extracted to define an abstract framework for exception-based update operators, viewing a knowledge base as the *set of sets of models* of its elements. It is shown that the framework can capture a wide range of both model- and formula-based classical update operators, and thus serves as the first firm formal ground connecting classical and rule updates.

1. Introduction

Recent standardisation efforts gave rise to widely accepted knowledge representation languages such as the Web Ontology Language (OWL)[1] and Rule Interchange Format (RIF),[2] based on Description Logics [8] and Logic Programming [14, 28, 45, 52], respectively. This has fostered a large number of ontologies and

---

rule bases with different levels of complexity and scale. Whereas ontologies provide the logical underpinning of intelligent access and information integration, rules are widely used to represent business policies, regulations and declarative guidelines about information.

Since both ontologies and rules offer important features for knowledge representation, considerable effort has been invested in identifying a unified hybrid knowledge framework where expressivity of both formalisms could be seamlessly combined. Over the years, work on hybrid knowledge bases has matured significantly and fundamental semantic as well as computational problems were addressed successfully [15, 16, 35, 44, 54]. While such formalisms make it possible to seamlessly combine rules and ontologies in a single unified framework, they do not take into account the *dynamic character* of application areas where they are to be used. More particularly, the essential support for keeping a hybrid knowledge base *up to date*, by incorporating new and possibly conflicting information, is still missing. Nonetheless, this topic has been extensively addressed in the context of both Description Logics and Logic Programs, when taken separately.

Ontology Updates

The area of research called *ontology change* encompasses a number of strongly related though distinguishable subareas, such as ontology matching, ontology integration and merging, or ontology translation [26]. The purest type of change, concerned with modifications to a single ontology, is generally referred to as *ontology evolution*. Approaches to ontology evolution with a firm semantic underpinning, thus amenable to a formal analysis of their behaviour and properties, are based on research in the area of *belief change*, initiated by the seminal work of Alchourrón, Gärdenfors and Makinson (AGM) [1] who proposed a set of desirable properties of change operators on monotonic logics, now referred to as the *AGM postulates*.

Subsequently, *revision* and *update* were distinguished as two very related but ultimately different belief change operations [41, 42, 73]. While revision deals with incorporating new information about a *static world* into a knowledge base, update takes place when a knowledge base needs to be brought up to date when the *modelled world changes*. While AGM postulates were deemed appropriate for describing revision, Katsuno and Mendelzon suggested a different set of postulates for updates: the *KM postulates for belief update* [41].

Update operators respecting the KM postulates, usually referred to as *model-based*, are based on the idea that the models of a knowledge base correspond to possible states of the represented world. When a change in the world needs to be recorded, inertia is applied to each of these possible states, making only the smallest necessary modifications to reflect the change, and arriving at a new collection of possible states that represent the world after the update. Since the updates are specified on the semantic level, they are naturally *syntax-independent*. These ideas, and particularly Winslett's update operator [42, 72], were later used to partially address ontology updates [9, 17–19, 23, 43, 51], namely to update the part of the ontology with assertions about individuals (the *ABox*).

On the other hand, model-based operators are considered inappropriate for updating ontological axioms that define the terminology (the *TBox*) [13, 64]. Their antipole, *formula-based operators*, which manipulate the knowledge base at a syntactic level and are strongly related to *base revision operators*, were adopted for performing TBox updates instead [13], and they also inspired a recent approach to ABox updates [49].

Rule Updates

When updates started to be investigated in the context of Logic Programming, it was only natural to adapt belief update principles and operators to this purpose [2, 53]. However, such approaches proved insufficiently expressive, principally because the model-based approach fails to capture the essential relationships between literals encoded in rules [48], and the formula-based approach is too crude as it does not

allow rules to be reactivated when reasons for their suppression disappear [75]. Although state-of-the-art approaches to rule updates are guided by the same basic intuitions and aspirations as belief updates, they build upon fundamentally different principles and methods.

Many of them are based on the *causal rejection principle* [3, 5, 24, 48, 55], which states that a rule is *rejected* only if it is directly contradicted by a more recent rule. This essentially means that inertia and minimal change is applied to rules instead of to the state, as with model-based belief updates. Causal rejection semantics are useful in a number of practical scenarios [4, 37, 59, 63] and their behaviour is intuitively predictable. Alternative approaches to rule updates employ syntactic transformations and other methods, such as abduction [60], prioritisation and preferences [20, 75], or dependencies on default assumptions [47, 62].

Despite the variety of techniques used in these approaches, certain properties are common to all of them. First, the stable models assigned to a program after one or more updates are always *supported*: for each true atom p there exists a rule in either the original program or its updates that has p in the head and whose body is satisfied. Second, all mentioned rule update semantics coincide when it comes to *updating sets of facts* by newer facts. We conjecture that any reasonable rule update semantics should indeed be in line with the basic intuitions regarding *support* and *fact update*. Another common characteristic of all these approaches is that they need to refer to the *syntactic structure* of a logic program: the individual rules and, in most cases, also the literals in their heads and bodies. This renders them seemingly irreconcilable with ontology updates since ontology axioms simply have no heads and bodies.

Towards Hybrid Updates

A unifying framework that could embrace both belief and rule updates is of great importance as it enables a deeper understanding of all involved methods and principles and creates room for their cross-fertilisation, ripening and further development. It is also important for the development of update semantics for hybrid knowledge bases – in [64, 70] we provided partial solutions to this problem but the inherent differences between the distinct approaches to updates have prevented us from suggesting a universal hybrid update semantics.

Moreover, we argue that syntax-independence, central to model-based belief updates, is essential and should be pursued at large in order to encourage a logical underpinning of all update operators and so facilitate analysis of their semantic properties. When equivalence with respect to classical models is inappropriate, as is the case with rules, syntax-independence should be retained by finding an appropriate notion of equivalence, specific to the underlying formalism and its use.

With these standpoints in mind, we proceed with our previous work addressing the logical foundations of rule updates. In [69] we have shown that *strong equivalence* is not a suitable basis for syntax-independent rule update operators because such operators cannot respect both support and fact update. This can be demonstrated on programs $\mathcal{P} = \{\, p., q. \,\}$ and $\mathcal{Q} = \{\, p., q \leftarrow p. \,\}$ which are strongly equivalent, so, due to syntax independence, an update asserting that p is now false ought to lead in both cases to the same stable models. Due to fact update, such an update on $\mathcal{P}$ should naturally lead to a stable model where q is true. But in case of $\mathcal{Q}$ such a stable model would be unsupported.

This led us to the study of stronger notions of program equivalence. In [65] we proposed to view a program as the *set of sets of models of its rules* in order to acknowledge rules as the atomic pieces of knowledge and, at the same time, abstract away from unimportant differences between their syntactic forms, focusing on their semantic content. In this paper we develop these ideas further and arrive at a unifying perspective on both classical and rule updates. More particularly, our main contributions are as follows:

- We introduce a novel monotonic characterisation of rules, *RE-models*, and show that they form a more suitable semantic foundation for rule updates than SE-models;

- We propose a generic scheme for defining semantic rule update operators: a program, viewed as the *set of sets of RE-models* of its rules, is updated by introducing additional interpretations to those sets of RE-models;

- We identify instances of the framework that bridge classical update principles with traditional rule update semantics: they combine *syntax-independence* with *support* and *fact update* and have other desirable *syntactic* as well as *semantic* properties.

- We solve the enduring problem of *state condensing* for two foundational rule update semantics by showing how they can be equivalently defined as binary operators on some class of logic programs;

- We define abstract exception-based operators for any knowledge representation formalism with a model-theoretic semantics;

- We show that exception-based operators capture a wide range of model- and formula-based belief update operators.

This paper is organised as follows: We introduce the necessary theoretical background in Section 2 and in Section 3 we define RE-models and associated notions of equivalence. Section 4 introduces the generic scheme for specifying exception-based rule update operators as well as its particular instances and analyses their theoretical properties. Section 5 is devoted to the problem of state condensing and introduces operators for condensing an update sequence into a single program for foundational rule update semantics. Sections 6 and 7 introduce abstract exception-based operators and show how they are able to characterise belief updates. We discuss our findings in a broader context and conclude in Section 8. Proofs of all formal results can be found in the appendices.

2. Background

2.1. Propositional Logic

We consider a propositional language over a finite set of propositional variables $\mathcal{A}$ and the usual set of propositional connectives to form propositional formulas. A *knowledge base* is a finite set of formulas. A (two-valued) interpretation is any $I \subseteq \mathcal{A}$ and the set of all interpretations is denoted by $\mathcal{I}$. The set of models of a knowledge base $\mathcal{B}$ is defined in the standard way and denoted by $[\![\mathcal{B}]\!]$. Given an interpretation I, we sometimes write $I \models \mathcal{B}$ or $I(\mathcal{B}) = \mathsf{T}$ if I is a model of $\mathcal{B}$, and $I \not\models \mathcal{B}$ or $I(\mathcal{B}) = \mathsf{F}$ if I is not a model of $\mathcal{B}$. A knowledge base $\mathcal{B}$ is *consistent* if $[\![\mathcal{B}]\!]$ is non-empty; *complete* if $[\![\mathcal{B}]\!]$ is a singleton set. Given two knowledge bases $\mathcal{B}, \mathcal{C}$, we say that $\mathcal{B}$ *entails* $\mathcal{C}$, denoted by $\mathcal{B} \models \mathcal{C}$, if $[\![\mathcal{B}]\!] \subseteq [\![\mathcal{C}]\!]$, and that $\mathcal{B}$ *is equivalent to* $\mathcal{C}$, denoted by $\mathcal{B} \equiv \mathcal{C}$, if $[\![\mathcal{B}]\!] = [\![\mathcal{C}]\!]$. The models, set of models, consistency, completeness, entailment and equivalence are generalised to formulas by treating every formula ϕ as the knowledge base $\{\phi\}$.

2.2. Logic Programs

The basic syntactic building blocks of rules are also propositional atoms from $\mathcal{A}$. A *default literal* is an atom preceded by $\sim$ denoting default negation. The set of all *literals* $\mathcal{L}$ consists of all atoms and default literals. The *complementary literal* to a literal L is denoted by $\overline{L}$ and defined as follows: for any atom p, $\overline{p} = \sim p$ and $\overline{\sim p} = p$. Given a set of literals S, we introduce the following notation: $S^+ = \{p \in \mathcal{A} \mid p \in S\}$, $S^- = \{p \in \mathcal{A} \mid \sim p \in S\}$, $\overline{S} = \{\overline{L} \mid L \in S\}$.

A *rule* is a pair of sets of literals $\pi = (\mathsf{H}_\pi, \mathsf{B}_\pi)$. We say that H_π is the *head of π* and B_π is the *body of π*. Usually, for convenience, we write π as $\left(\mathsf{H}_\pi^+; \overline{\mathsf{H}_\pi^-} \leftarrow \mathsf{B}_\pi^+, \overline{\mathsf{B}_\pi^-}.\right)$. A rule is called *non-disjunctive* if its head contains at most one literal; *a fact* if its head contains exactly one literal and its body is empty. A *program* is any set of rules. A program is *non-disjunctive* if all its rules are; *acyclic* if it satisfies the conditions set out in [6]. We also introduce the following non-standard notion that is needed throughout the rest of the paper:

Definition 1 (Canonical Tautology). Let $p_\tau \in \mathcal{A}$ be fixed. The *canonical tautology* τ is the rule $(p_\tau \leftarrow p_\tau.)$.

Turning to the semantics, we need to define *stable models* [28, 29] and *SE-models* [56, 71] of a logic program. We start by generalizing the set of *(classical) models* to literals, rules and programs. Given an atom p, a set of literals S, a rule π and a program $\mathcal{P}$, we define the following:

$$\llbracket \sim p \rrbracket = \mathfrak{I} \setminus \llbracket p \rrbracket \;, \qquad\qquad \llbracket S \rrbracket = \bigcap_{L \in S} \llbracket L \rrbracket \;,$$
$$\llbracket \pi \rrbracket = \{\, J \in \mathfrak{I} \mid J \notin \llbracket \mathsf{B}_\pi \rrbracket \vee \exists L \in \mathsf{H}_\pi : J \in \llbracket L \rrbracket \,\} \;, \qquad\qquad \llbracket \mathcal{P} \rrbracket = \bigcap_{\pi \in \mathcal{P}} \llbracket \pi \rrbracket \;.$$

For a set of literals S, we also write $J \models S$ whenever $J \in \llbracket S \rrbracket$. A program $\mathcal{P}$ is *consistent* if $\llbracket \mathcal{P} \rrbracket \neq \emptyset$.

The stable and SE-models are defined in terms of reducts. Given a rule π and an interpretation J, the *reduct of π w.r.t. J*, denoted by π^J, is the rule $(\mathsf{H}_\pi^+ \leftarrow \mathsf{B}_\pi^+.)$ if $J \models \overline{\mathsf{B}_\pi^-}$ and $J \models \mathsf{H}_\pi^-$; otherwise it is the canonical tautology τ.[3] The *reduct of a program $\mathcal{P}$ w.r.t. J* is defined as $\mathcal{P}^J = \{\, \pi^J \mid \pi \in \mathcal{P} \,\}$.

An interpretation J is a *stable model* of a program $\mathcal{P}$ if J is a subset-minimal model of $\mathcal{P}^J$. The set of all stable models of $\mathcal{P}$ is denoted by $\llbracket \mathcal{P} \rrbracket_{\mathsf{SM}}$.

SE-models are semantic structures that can be seen as three-valued interpretations. In particular, we call a pair of interpretations $X = (I, J)$ such that $I \subseteq J$ a *three-valued interpretation*. Each atom p is assigned one of three truth values in X: $X(p) = \mathsf{T}$ if $p \in I$; $X(p) = \mathsf{U}$ if $p \in J \setminus I$; $X(p) = \mathsf{F}$ if $p \in \mathcal{A} \setminus J$. The set of all three-valued interpretations is denoted by $\mathfrak{X}$. A three-valued interpretation (I, J) is an *SE-model* of a rule π if J is a model of π and I is a model of π^J. The set of all SE-models of a rule π is denoted by $\llbracket \pi \rrbracket_{\mathsf{SE}}$ and for any program $\mathcal{P}$, $\llbracket \mathcal{P} \rrbracket_{\mathsf{SE}} = \bigcap_{\pi \in \mathcal{P}} \llbracket \pi \rrbracket_{\mathsf{SE}}$. Note that J is a stable model of $\mathcal{P}$ if and only if $(J, J) \in \llbracket \mathcal{P} \rrbracket_{\mathsf{SE}}$ and for all $I \subsetneq J$, $(I, J) \notin \llbracket \mathcal{P} \rrbracket_{\mathsf{SE}}$. Also, $J \in \llbracket \mathcal{P} \rrbracket$ if and only if $(J, J) \in \llbracket \mathcal{P} \rrbracket_{\mathsf{SE}}$. We say that a rule π is **(SE-)tautological** if $\llbracket \pi \rrbracket_{\mathsf{SE}} = \mathfrak{X}$. Note that the canonical tautology (c.f. Definition 1) is tautological.

2.3. Belief Updates

In this section we briefly introduce *model-based* as well as *formula-based* belief update operators which form the basis of formal approaches to ontology updates [9, 13, 17–19, 23, 43, 49, 51].

We liberally define a belief update operator as any function that takes the original knowledge base and its update as inputs and returns the updated knowledge base.

Definition 2 (Update Operator). A *belief update operator* is a binary function over the set of all knowledge bases. Any belief update operator $\diamond$ is inductively generalised to finite sequences of knowledge bases $\langle \mathcal{B}_i \rangle_{i<n}$ as follows: $\diamond \langle \rangle = \emptyset$ and $\diamond \langle \mathcal{B}_i \rangle_{i<n+1} = (\diamond \langle \mathcal{B}_i \rangle_{i<n}) \diamond \mathcal{B}_n$.

---

[3] In other words, if J satisfies all default literals in the body of the rule and none of the default literals in its head, then the reduct is obtained from the original rule by removing all default literals from its head and body. In other cases, J is certainly a model of the rule, so the reduct is defined as the canonical tautology.

Table 1: Belief update postulates

Postulate	Knowledge base as a single formula	Knowledge base as a set
(B1)	$\phi \diamond \mu \models \mu$.	$\mathcal{B} \diamond \mathcal{U} \models \mathcal{U}$.
(B2)	If $\phi \models \mu$, then $\phi \diamond \mu \equiv \phi$.	If $\mathcal{B} \models \mathcal{U}$, then $\mathcal{B} \diamond \mathcal{U} \equiv \mathcal{B}$.
(B2.$\top$)	$\phi \diamond \top \equiv \phi$.	$\mathcal{B} \diamond \emptyset \equiv \mathcal{B}$.
(B2.1)	$\phi \wedge \mu \models \phi \diamond \mu$.	$\mathcal{B} \wedge \mathcal{U} \models \mathcal{B} \diamond \mathcal{U}$.
(B2.2)	$(\phi \wedge \mu) \diamond \mu \models \phi$.	$(\mathcal{B} \wedge \mathcal{U}) \diamond \mathcal{U} \models \mathcal{B}$.
(B3)	If $[\![\phi]\!] \neq \emptyset$ and $[\![\mu]\!] \neq \emptyset$, then $[\![\phi \diamond \mu]\!] \neq \emptyset$.	If $[\![\mathcal{B}]\!] \neq \emptyset$ and $[\![\mathcal{U}]\!] \neq \emptyset$, then $[\![\mathcal{B} \diamond \mathcal{U}]\!] \neq \emptyset$.
(B4)	If $\phi \equiv \psi$ and $\mu \equiv \nu$, then $\phi \diamond \mu \equiv \psi \diamond \nu$.	If $\mathcal{B} \equiv \mathcal{C}$ and $\mathcal{U} \equiv \mathcal{V}$, then $\mathcal{B} \diamond \mathcal{U} \equiv \mathcal{C} \diamond \mathcal{V}$.
(B5)	$(\phi \diamond \mu) \wedge \nu \models \phi \diamond (\mu \wedge \nu)$.	$(\mathcal{B} \diamond \mathcal{U}) \cup \mathcal{V} \models \mathcal{B} \diamond (\mathcal{U} \cup \mathcal{V})$.
(B6)	If $\phi \diamond \mu \models \nu$ and $\phi \diamond \nu \models \mu$, then $\phi \diamond \mu \equiv \phi \diamond \nu$.	If $\mathcal{B} \diamond \mathcal{U} \models \mathcal{V}$ and $\mathcal{B} \diamond \mathcal{V} \models \mathcal{U}$, then $\mathcal{B} \diamond \mathcal{U} \equiv \mathcal{B} \diamond \mathcal{V}$.
(B7)	If ϕ is complete, then $(\phi \diamond \mu) \wedge (\phi \diamond \nu) \models \phi \diamond (\mu \vee \nu)$.	
(B8)	$(\phi \vee \psi) \diamond \mu \equiv (\phi \diamond \mu) \vee (\psi \diamond \mu)$.	

The fundamental idea underlying model-based update operators [41, 42, 72, 73] is that models of the original knowledge base are viewed as *possible* or *admissible states* of the modelled domain, only one of which is the actual state. Given this perspective, it is natural to perform an update with $\mathcal{U}$ by modifying each of the possible states independently of the others, making it consistent with $\mathcal{U}$, and thus obtaining a new set of interpretations – the models of the updated knowledge base. Formally this is captured by the equation

$$[\![\mathcal{B} \diamond \mathcal{U}]\!] = \bigcup_{I \in [\![\mathcal{B}]\!]} \mathsf{incorporate}([\![\mathcal{U}]\!], I) \ , \tag{1}$$

where $\mathsf{incorporate}(\mathcal{M}, I)$ returns the members of $\mathcal{M}$ closer to I so that the original information in I is preserved as much as possible. For instance, Winslett's operator, first introduced to deal with change in action theories and relational databases with incomplete information [42, 72], and used extensively to study ontology updates [9, 17–19, 23, 43, 51], is characterised by equation (1) where $\mathsf{incorporate}(\mathcal{M}, I)$ is the set of models from $\mathcal{M}$ that interpret a minimal set of atoms differently than I. From a more general perspective, Katsuno and Mendelzon have shown [41] that if each knowledge base is represented by a single formula and $\mathsf{incorporate}(\mathcal{M}, I)$ is bound to return those members of $\mathcal{M}$ that are minimal w.r.t. a closeness relation $\leq_I$ assigned to I,[4] then the class of operators which satisfy (1) coincides with the class of operators which satisfy the declarative postulates (B1) – (B8) listed in Table 1. Note that (B1) – (B6) can be immediately generalised to deal with arbitrary knowledge bases but (B7) and (B8) require disjunction of knowledge bases to be defined.

Some of these postulates, and especially (B4) which guarantees syntax-independence of an update operator, continue to be seen as fundamental cornerstones of belief updates [34]. The desirability of others has been questioned by many [10, 11, 22, 34], and particularly (B2), (B5), (B6) and (B7) are deemed controversial: the first three sometimes lead to undesirable behaviour while the last one is hard to explain intuitively and is satisfied only by a minority of existing update operators [34]. Note also that (B2) entails the weaker

---

[4]Technically, the closeness relation is simply a partial order on $\mathfrak{I}$ such that I is the least interpretation w.r.t. $\leq_I$.

principles (B2.⊤), (B2.1) and (B2.2). The first two are uncontroversial as they are satisfied by all update operators. In addition, in the presence of (B4), the latter two are together powerful enough to entail (B2), so the controversial part of (B2) is (B2.2) [34].

Earlier approaches to updates, dubbed *formula-based* [73], operate on the syntax of a knowledge base and, as a consequence, are not syntax-independent. Nevertheless, they have recently been considered for performing ontology updates [13, 49]. Traditional formula-based update operators are Set-Of-Theories [25], WIDTIO (When In Doubt, Throw It Out) [30, 31, 73] and Cross-Product [30]. We define only the latter two as the Set-Of-Theories operator produces a *collection of knowledge bases* as its result instead of a single knowledge base, and is equivalent to the Cross-Product operator which compiles these knowledge bases into one.

The central notion in these operators is that of a *possible remainder* which is a maximal set of formulas from the original knowledge base that is consistent with the update. Formally, given knowledge bases $\mathcal{B}$ and $\mathcal{U}$, the set of possible remainders $\mathrm{rem}(\mathcal{B},\mathcal{U})$ is the set of maximal subsets $\mathcal{B}'$ of $\mathcal{B}$ such that $\mathcal{B}' \cup \mathcal{U}$ is consistent. The distinct formula-based operators differ in how they deal with the case when there is more than one possible remainder. While Cross-Product compiles the different remainders into a single formula, WIDTIO takes the safer path – it keeps exactly those formulas that belong to the intersection of all remainders and throws away the rest. Additionally, in [13] the new formula-based operator *Bold* was suggested for performing TBox updates. The Bold operator solves the problem of multiple remainders by using a selection function to choose one and commit to it. Formally, a *remainder selection function* is a function s that assigns to every set of remainders $\mathcal{R}$ a remainder $s(\mathcal{R}) \in \mathcal{R}$. Given such a selection function s, the Cross-Product operator $\circ_{\mathrm{cp}}$, WIDTIO operator $\circ_{\mathrm{widtio}}$ and Bold operator $\circ_{\mathrm{bold}}^{s}$ are defined for all knowledge bases $\mathcal{B}, \mathcal{U}$ as follows:

$$\mathcal{B} \circ_{\mathrm{cp}} \mathcal{U} = \mathcal{U} \cup \left\{ \bigvee_{\mathcal{B}' \in \mathrm{rem}(\mathcal{B},\mathcal{U})} \bigwedge \mathcal{B}' \right\}, \quad \mathcal{B} \circ_{\mathrm{widtio}} \mathcal{U} = \mathcal{U} \cup \bigcap \mathrm{rem}(\mathcal{B},\mathcal{U}), \quad \mathcal{B} \circ_{\mathrm{bold}}^{s} \mathcal{U} = \mathcal{U} \cup s(\mathrm{rem}(\mathcal{B},\mathcal{U})).$$

2.4. Rule Updates

Rule update semantics assign stable models to pairs or sequences of programs where each component represents an update of the preceding ones. In the following, we formalise some of the intuitions behind these semantics and define two foundational rule update semantics.

We start with the basic concepts. A *dynamic logic program* (DLP) is a finite sequence of non-disjunctive programs. Given a DLP $\boldsymbol{P}$, we use $\mathrm{all}(\boldsymbol{P})$ to denote the set of all rules belonging to the programs in $\boldsymbol{P}$.[5] We say that $\boldsymbol{P}$ is *acyclic* if $\mathrm{all}(\boldsymbol{P})$ is acyclic. A rule update semantics S assigns a set of *S-stable models* $[\![\boldsymbol{P}]\!]_{\mathsf{S}}$ to every DLP $\boldsymbol{P}$.

As indicated in the introduction, rule update semantics implicitly follow certain basic intuitions. Particularly, they produce *supported models* and their behaviour coincides when it comes to *updating sets of facts* by newer facts. We formalise these two properties w.r.t. rule update semantics for DLPs, calling them *syntactic* because their formulation requires that we refer to the syntax of the respective DLP.

In the static setting, support [7, 21] is one of the basic conditions that Logic Programming semantics are intuitively designed to satisfy. Its generalisation to the dynamic case is straightforward.

Syntactic Property 1 (Support)**.** *Let $\mathcal{P}$ be a program, p an atom and J an interpretation. We say that $\mathcal{P}$ supports p in J if $p \in \mathsf{H}_{\pi}$ and $J \models \mathsf{B}_{\pi}$ for some rule $\pi \in \mathcal{P}$.*

---

[5]In order to avoid issues with rules that are repeated in multiple components of a DLP, we assume throughout this paper that every rule is uniquely identified in all set-theoretic operations. This can be formalised by assigning a unique name to each rule and performing operations on names instead of the rules themselves. However, for the sake of simplicity, we leave the technical realisation to the reader.

A rule update semantics S *respects support* if for every DLP $\boldsymbol{P}$ and every S-stable model J of $\boldsymbol{P}$ the following condition is satisfied: Every atom $p \in J$ is supported by $\mathsf{all}(\boldsymbol{P})$ in J.

Thus, if a rule update semantics S respects support, then there is at least *some* justification for every atom in an assigned S-stable model.

The second syntactic property that is generally adhered to is the usual expectation regarding how facts are to be updated by newer facts. It enforces a limited notion of state inertia but only for the case when both the initial program and its updates are consistent sets of facts.

Syntactic Property 2 (Fact Update)**.** *A rule update semantics* S *respects fact update* if for every finite sequence of consistent sets of facts $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$, the unique S-stable model of $\boldsymbol{P}$ is the interpretation

$$\{\, p \mid \exists i < n : (p.) \in \mathcal{P}_i \wedge (\forall j : i < j < n \Rightarrow ({\sim} p.) \notin \mathcal{P}_j) \,\} \quad .$$

We also introduce two further syntactic properties that are more tightly bound to approaches based on the causal rejection principle [3, 5, 24, 48, 55]. The first one states the principle itself, under the assumption that a *conflict* between rules occurs if and only if the rules have complementary heads.

Syntactic Property 3 (Causal Rejection)**.** *A rule update semantics* S *respects causal rejection* if for every DLP $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$, every S-stable model J of $\boldsymbol{P}$, all $i < n$ and all rules $\pi \in \mathcal{P}_i$, if J is not a model of π, then there exists a rule $\sigma \in \mathcal{P}_j$ with $j > i$ such that $\mathsf{H}_\sigma = \overline{\mathsf{H}_\pi}$ and $J \models \mathsf{B}_\sigma$.

Intuitively, the principle requires that all *rejected* rules, i.e. rules that are not satisfied in an S-stable model J, must be in conflict with a more recent rule whose body is satisfied in J. This rule then provides a *cause* for the rejection.

The final syntactic property stems from the fact that all rule update semantics based on causal rejection coincide on acyclic DLPs [5, 36]. Thus, the behaviour of any rule update semantics on acyclic DLPs can be used as a way to compare it to all these semantics simultaneously. Before formalising the property, we define two foundational rule update semantics based on causal rejection: the JU-semantics [48] and the UA-semantics [24].

Definition 3 (JU-Semantics [48] and UA-Semantics [24])**.** Let $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ be a DLP and J an interpretation. The sets of rejected rules $\mathsf{rej}_{\mathsf{JU}}(\boldsymbol{P}, J)$ and $\mathsf{rej}_{\mathsf{UA}}(\boldsymbol{P}, J)$ are defined as follows:

$$\mathsf{rej}_{\mathsf{JU}}(\boldsymbol{P}, J) = \big\{\, \pi \in \mathcal{P}_i \mid \exists j \, \exists \sigma \in \mathcal{P}_j : i < j < n \wedge \mathsf{H}_\sigma = \overline{\mathsf{H}_\pi} \wedge J \models \mathsf{B}_\sigma \,\big\} \ ,$$
$$\mathsf{rej}_{\mathsf{UA}}(\boldsymbol{P}, J) = \big\{\, \pi \in \mathcal{P}_i \mid \exists j \, \exists \sigma \in \mathcal{P}_j \setminus \mathsf{rej}_{\mathsf{UA}}(\boldsymbol{P}, J) : i < j < n \wedge \mathsf{H}_\sigma = \overline{\mathsf{H}_\pi} \wedge J \models \mathsf{B}_\sigma \,\big\} \ .^6$$

The set $[\![\boldsymbol{P}]\!]_{\mathsf{JU}}$ of *JU-models of a DLP* $\boldsymbol{P}$ consists of all interpretations J such that J is a stable model of the program $\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}_{\mathsf{JU}}(\boldsymbol{P}, J)$. Similarly, the set $[\![\boldsymbol{P}]\!]_{\mathsf{UA}}$ of *UA-models of a DLP* $\boldsymbol{P}$ consists of all interpretations J such that J is a stable model of the program $\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}_{\mathsf{UA}}(\boldsymbol{P}, J)$.

Under the JU-semantics, a rule π is rejected if a more recent rule σ is in conflict with π and the body of σ is satisfied in the stable model candidate J. The only difference in the UA-semantics [24] is that rejected rules are prevented from rejecting other rules. Perhaps surprisingly, this renders the UA-semantics more sensitive to tautological updates, which cannot indicate any change in the modelled world, than the JU-semantics. For example, the DLP

$$\boldsymbol{P}_1 = \langle \{\, p. \,\}, \{\, {\sim} p. \,\}, \{\, p \leftarrow p. \,\} \rangle \tag{2}$$

8

has only one JU-model, $\emptyset$, but the UA-semantics admits the additional undesired model $\{\,p\,\}$. Nevertheless, there are also situations in which the JU-semantics assigns additional models only due to the presence of tautological rules. This is discussed in more detail in Section 8.

The final syntactic property can now be stated as follows:

Syntactic Property 4 (Acyclic Justified Update). *A rule update semantics* S *respects acyclic justified update if for every acyclic DLP* $\boldsymbol{P}$, *the set of* S*-stable models of* $\boldsymbol{P}$ *is* $[\![\boldsymbol{P}]\!]_{\text{JU}}$.

2.5. Program Equivalence

While in propositional logic equivalence under classical models is *the* equivalence, there is no such single notion of program equivalence. When considering Answer-Set Programs, the first choice is *stable equivalence* (or SM*-equivalence*) that compares programs based on their sets of stable models.

In many cases, however, SM-equivalence is not strong enough because programs with the same stable models, when augmented with the same additional rules, may end up having completely different stable models. This gives rise to the notion of *strong equivalence* [50] which requires that stable models stay the same even in the presence of additional rules. It is a well-known fact that programs are strongly equivalent if and only if they have the same set of SE-models [71]. Thus, we refer to strong equivalence as SE*-equivalence*.

But even SE-equivalence is not satisfactory when used as a basis for syntax-independent rule update operators because such operators cannot respect both support and fact update [69]. So in order to arrive at syntax-independent rule update operators that satisfy the basic intuitions underlying rule updates, we need to search for a notion of program equivalence that is stronger than SE-equivalence. One candidate is the *strong update equivalence* (or SU*-equivalence*) [39], which requires that under both additions and removals of rules, stable models of the two programs in question remain the same. It has been shown in [39] that this notion of equivalence is very strong – programs are SU-equivalent only if they contain exactly the same non-tautological rules, and in addition, each of them may contain some tautological ones. Thus, this notion of program equivalence seems perhaps *too strong* as it is not difficult to find rules such as $(\sim p \leftarrow p.)$ and $(\leftarrow p.)$ that are syntactically different but carry the same meaning.

This observation resulted in the definition of *strong rule equivalence* (or SR*-equivalence*) and *strong minimal rule equivalence* (or SMR*-equivalence*) in [65] that, in terms of strength, fall between SE-equivalence and SU-equivalence. It is based on the idea of viewing a program $\mathcal{P}$ as the set of sets of SE-models of its rules $\langle\!\langle \mathcal{P} \rangle\!\rangle_{\text{SE}} = \{\, [\![\pi]\!]_{\text{SE}} \mid \pi \in \mathcal{P} \,\}$.

The five mentioned notions of program equivalence are defined as follows:

Definition 4 (Program Equivalence). Let $\mathcal{P}, \mathcal{Q}$ be programs, $\mathcal{P}^\tau = \mathcal{P} \cup \{\,\tau\,\}$, $\mathcal{Q}^\tau = \mathcal{Q} \cup \{\,\tau\,\}$, let $\min \mathcal{S}$ denote the subset-minimal elements of any set of sets $\mathcal{S}$ and $\div$ denote set-theoretic symmetric difference. We write

$$\mathcal{P} \equiv_{\text{SM}} \mathcal{Q} \quad \text{whenever} \quad [\![\mathcal{P}]\!]_{\text{SM}} = [\![\mathcal{Q}]\!]_{\text{SM}} \;; \quad \mathcal{P} \equiv_{\text{SMR}} \mathcal{Q} \quad \text{whenever} \quad \min\langle\!\langle \mathcal{P}^\tau \rangle\!\rangle_{\text{SE}} = \min\langle\!\langle \mathcal{Q}^\tau \rangle\!\rangle_{\text{SE}} \;;$$

$$\mathcal{P} \equiv_{\text{SE}} \mathcal{Q} \quad \text{whenever} \quad [\![\mathcal{P}]\!]_{\text{SE}} = [\![\mathcal{Q}]\!]_{\text{SE}} \;; \quad \mathcal{P} \equiv_{\text{SR}} \mathcal{Q} \quad \text{whenever} \quad \langle\!\langle \mathcal{P}^\tau \rangle\!\rangle_{\text{SE}} = \langle\!\langle \mathcal{Q}^\tau \rangle\!\rangle_{\text{SE}} \;;$$

$$\mathcal{P} \equiv_{\text{SU}} \mathcal{Q} \quad \text{whenever} \quad [\![\mathcal{P} \div \mathcal{Q}]\!]_{\text{SE}} = \mathcal{X} \;.$$

We say that $\mathcal{P}$ *is* X*-equivalent to* $\mathcal{Q}$ if $\mathcal{P} \equiv_{\text{X}} \mathcal{Q}$.

So two programs are SR-equivalent if they contain the same rules, modulo SE-models; τ is added to both programs so that presence or absence of tautological rules in a program does not influence program equivalence. In the case of SMR-equivalence, only the subset-minimal sets of SE-models are compared,

the motivation being that programs such as $\mathcal{P} = \{\, p \leftarrow q. \,\}$ and $\mathcal{Q} = \{\, p., p \leftarrow q. \,\}$, when updated, should behave the same way since the extra rule in $\mathcal{Q}$ is just a weakened version of the rule in $\mathcal{P}$. Though $\mathcal{P}$ and $\mathcal{Q}$ are not SR-equivalent, they are SMR-equivalent.

To formally capture the comparison of strength between these notions of program equivalence, we write $\equiv_X \preceq \equiv_Y$ if $\mathcal{P} \equiv_Y \mathcal{Q}$ implies $\mathcal{P} \equiv_X \mathcal{Q}$ and $\equiv_X \prec \equiv_Y$ if $\equiv_X \preceq \equiv_Y$ but not $\equiv_Y \preceq \equiv_X$. Then:

Proposition 5 ([65]). $\equiv_{SM} \prec \equiv_{SE} \prec \equiv_{SMR} \prec \equiv_{SR} \prec \equiv_{SU}$.

3. Robust Equivalence Models

In [65] we studied the expressivity of SE-models with respect to a single rule. On the one hand, SE-models turned out to be a useful means of stripping away irrelevant syntactic details. On the other hand, a rule with a default literal in its head is indistinguishable from an integrity constraint [12, 38, 40]. For example, the rules

$$\leftarrow p, q. \qquad\qquad \sim p \leftarrow q. \qquad\qquad \sim q \leftarrow p. \qquad\qquad (3)$$

have the same set of SE-models. In a static setting, these rules indeed carry essentially the same meaning: "it must not be the case that p and q are both true". But in a dynamic context, the latter two rules may, in addition, express that the truth of one atom gives a *reason* for the other atom to *cease being true*. For example, an update of the program $\{\, p., q. \,\}$ by $\{\, \sim p \leftarrow q. \,\}$ leads to the stable model $\{\, q \,\}$ while an update by $\{\, \sim q \leftarrow p. \,\}$ to the stable model $\{\, p \,\}$. This convention is adopted by causal rejection-based rule update semantics [3, 5, 24, 48, 55] which constitute one of the most mature approaches to rule updates.

In order to be able to semantically characterise causal rejection-based rule update semantics, we need to distinguish between constraints and rules with default literals in their heads. These classes can be formally captured as follows:

Definition 6 (Constraint and Abolishing Rule). A rule π is a *constraint* if $\mathsf{H}_\pi = \emptyset$ and B_π^+ is disjoint with B_π^-.[7] A rule π is *abolishing* if $\mathsf{H}_\pi^+ = \emptyset$, $\mathsf{H}_\pi^- \neq \emptyset$ and the sets H_π^-, B_π^+ and B_π^- are pairwise disjoint.

So what we are looking for is a semantic characterisation of rules that

1) can distinguish constraints from related abolishing rules;
2) discards irrelevant syntactic details (akin to SE-models);
3) has a clear link to stable models (akin to SE-models).

In the following we introduce a novel monotonic semantics that exactly meets these criteria. We show that it possesses the desired properties and use it to introduce a notion of program equivalence that is strong enough as a basis for syntax-independent rule update operators.

Without further ado, *robust equivalence models*, or *RE-models* for short, are defined as follows:

Definition 7 (RE-Model). Let π be a rule. A three-valued interpretation $(I, J) \in \mathcal{X}$ is an *RE-model of π* if $I \models \pi^J$. The set of all RE-models of a rule π is denoted by $[\![\pi]\!]_{RE}$ and for any program $\mathcal{P}$, $[\![\mathcal{P}]\!]_{RE} = \bigcap_{\pi \in \mathcal{P}} [\![\pi]\!]_{RE}$.

We say that a rule π is *RE-tautological* if $[\![\pi]\!]_{RE} = \mathcal{X}$. Rules π, σ are *RE-equivalent* if $[\![\pi]\!]_{RE} = [\![\sigma]\!]_{RE}$.

---

[7]The latter condition guarantees that a constraint is not tautological.

Thus, unlike with SE-models, it is not required that $J \models \pi$ in order for (I, J) to be an RE-model of π. As a consequence, RE-models can distinguish between rules in (3): while both $(\{q\}, \{p, q\})$ and $(\{p\}, \{p, q\})$ are RE-models of the constraint, the former is not an RE-model of the first abolishing rule and the latter is not an RE-model of the second abolishing rule. This property holds in general, establishing requirement 1):

Proposition 8. *If π, σ are two different abolishing rules or an abolishing rule and a constraint, then π, σ are not RE-equivalent.*

Proof. See Appendix A, page 34. □

As for requirement 2), we first note that RE-equivalence is a refinement of SE-equivalence – there are no rules that are RE-equivalent but not SE-equivalent. The following result also shows that it is *only* the ability to distinguish between constraints and abolishing rules that is introduced by RE-models – rules that are not RE-equivalent to abolishing rules are distinguishable by RE-models if and only if they are distinguishable by SE-models. Furthermore, the classes of SE-tautological and RE-tautological rules coincide, so we can simply use the word *tautological* without ambiguity.

Proposition 9 (RE-Equivalence vs. SE-Equivalence)**.**
- *If two rules are RE-equivalent, then they are SE-equivalent.*
- *If two rules, neither of which is RE-equivalent to an abolishing rule, are SE-equivalent, then they are RE-equivalent.*
- *A rule is RE-tautological if and only if it is SE-tautological.*

Proof. See Appendix A, page 35. □

The affinity between SE-models and stable models is fully retained by RE-models, which establishes requirement 3).

Proposition 10 (RE-Models vs. Stable Models)**.** *An interpretation J is a stable model of a program $\mathcal{P}$ if and only if $(J, J) \in [\![\mathcal{P}]\!]_{\mathsf{RE}}$ and for all $I \subsetneq J$, $(I, J) \notin [\![\mathcal{P}]\!]_{\mathsf{RE}}$.*

Proof. See Appendix A, page 35. □

Also worth noting is that any set of three-valued interpretations can be expressed by a program using RE-models. This is not the case with SE-models since only well-defined sets of three-valued interpretations have corresponding programs.

Proposition 11. *For any $\mathcal{M} \subseteq \mathcal{X}$ there exists a program $\mathcal{P}$ such that $[\![\mathcal{P}]\!]_{\mathsf{RE}} = \mathcal{M}$.*

Proof. See Appendix A, page 35. □

Further properties of RE-models, analogous to those established in [65] for SE-models, can be found in Appendix A starting on page 31.

Since RE-models are able to distinguish constraints from abolishing rules while keeping the essential properties of SE-models, we henceforth adopt them as the basis for defining syntax-independent rule update operators. We denote the set of sets of RE-models of rules inside a program $\mathcal{P}$ by $\langle\!\langle \mathcal{P} \rangle\!\rangle_{\mathsf{RE}} = \{ [\![\pi]\!]_{\mathsf{RE}} \mid \pi \in \mathcal{P} \}$. We also introduce three additional notions of program equivalence: RE-, RR- and RMR-equivalence that are analogous to SE-, SR- and SMR-equivalence.
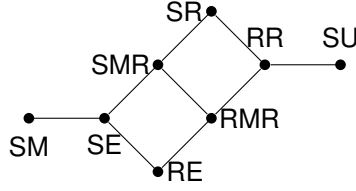
Figure 1: Notions of program equivalence and entailment from the weakest on the left side to the strongest on the right side. A missing link between X and Y indicates that $\equiv_X$ is incomparable with $\equiv_Y$ and $\models_X$ is incomparable with $\models_Y$, i.e. *none* of the following four relations holds: $\equiv_X \preceq \equiv_Y$, $\equiv_Y \preceq \equiv_X$, $\models_X \preceq \models_Y$, $\models_Y \preceq \models_X$.

Definition 12 (Program Equivalence Using RE-Models). Let $\mathcal{P}, \mathcal{Q}$ be programs, $\mathcal{P}^\tau = \mathcal{P} \cup \{\tau\}$, $\mathcal{Q}^\tau = \mathcal{Q} \cup \{\tau\}$. We write

$$\mathcal{P} \equiv_{\mathsf{RE}} \mathcal{Q} \quad \text{whenever} \quad [\![\mathcal{P}]\!]_{\mathsf{RE}} = [\![\mathcal{Q}]\!]_{\mathsf{RE}} \; ; \quad \mathcal{P} \equiv_{\mathsf{RMR}} \mathcal{Q} \quad \text{whenever} \quad \min\langle\!\langle \mathcal{P}^\tau \rangle\!\rangle_{\mathsf{RE}} = \min\langle\!\langle \mathcal{Q}^\tau \rangle\!\rangle_{\mathsf{RE}} \; ;$$

$$\mathcal{P} \equiv_{\mathsf{RR}} \mathcal{Q} \quad \text{whenever} \quad \langle\!\langle \mathcal{P}^\tau \rangle\!\rangle_{\mathsf{RE}} = \langle\!\langle \mathcal{Q}^\tau \rangle\!\rangle_{\mathsf{RE}} \; .$$

In order to consider belief update principles in the context of rule updates, we also need to establish notions of *program entailment* which are in line with the above defined program equivalence relations. This task is troublesome in case of SM-equivalence because the usage of entailment in belief update postulates is clearly a monotonic one while stable models are non-monotonic. For instance, a reformulation of (B1) would require that $\mathcal{P} \oplus \mathcal{U} \models \mathcal{U}$, though there is no reason for $\mathcal{P} \oplus \mathcal{U}$ to have *less* stable models than (or the same as) $\mathcal{U}$. Due to these issues, we refrain from defining SM-entailment. The remaining entailment relations are defined as follows:

Definition 13 (Program Entailment). Let $\mathcal{P}, \mathcal{Q}$ be programs, $\mathcal{P}^\tau = \mathcal{P} \cup \{\tau\}$ and $\mathcal{Q}^\tau = \mathcal{Q} \cup \{\tau\}$. We write

$$\mathcal{P} \models_{\mathsf{SE}} \mathcal{Q} \quad \text{whenever} \quad [\![\mathcal{P}]\!]_{\mathsf{SE}} \subseteq [\![\mathcal{Q}]\!]_{\mathsf{SE}} \; ; \quad \mathcal{P} \models_{\mathsf{SMR}} \mathcal{Q} \quad \text{whenever} \quad \forall \sigma \in \mathcal{Q}^\tau \, \exists \pi \in \mathcal{P}^\tau : [\![\pi]\!]_{\mathsf{SE}} \subseteq [\![\sigma]\!]_{\mathsf{SE}} \; ;$$

$$\mathcal{P} \models_{\mathsf{RE}} \mathcal{Q} \quad \text{whenever} \quad [\![\mathcal{P}]\!]_{\mathsf{RE}} \subseteq [\![\mathcal{Q}]\!]_{\mathsf{RE}} \; ; \quad \mathcal{P} \models_{\mathsf{RMR}} \mathcal{Q} \quad \text{whenever} \quad \forall \sigma \in \mathcal{Q}^\tau \, \exists \pi \in \mathcal{P}^\tau : [\![\pi]\!]_{\mathsf{RE}} \subseteq [\![\sigma]\!]_{\mathsf{RE}} \; ;$$

$$\mathcal{P} \models_{\mathsf{SU}} \mathcal{Q} \quad \text{whenever} \quad [\![\mathcal{Q} \setminus \mathcal{P}]\!]_{\mathsf{SE}} = \mathcal{X} \; ; \quad \mathcal{P} \models_{\mathsf{SR}} \mathcal{Q} \quad \text{whenever} \quad \forall \sigma \in \mathcal{Q}^\tau \, \exists \pi \in \mathcal{P}^\tau : [\![\pi]\!]_{\mathsf{SE}} = [\![\sigma]\!]_{\mathsf{SE}} \; ;$$

$$\mathcal{P} \models_{\mathsf{RR}} \mathcal{Q} \quad \text{whenever} \quad \forall \sigma \in \mathcal{Q}^\tau \, \exists \pi \in \mathcal{P}^\tau : [\![\pi]\!]_{\mathsf{RE}} = [\![\sigma]\!]_{\mathsf{RE}} \; .$$

We say that $\mathcal{P}$ *X-entails* $\mathcal{Q}$ if $\mathcal{P} \models_X \mathcal{Q}$.

As the following proposition shows, the defined entailment relations are fully in line with the corresponding equivalence relations.

Proposition 14. *Let X be one of SE, RE, SMR, RMR, SR, RR, SU and $\mathcal{P}, \mathcal{Q}$ be programs. Then, $\mathcal{P} \equiv_X \mathcal{Q}$ if and only if $\mathcal{P} \models_X \mathcal{Q}$ and $\mathcal{Q} \models_X \mathcal{P}$.*

Proof. See Appendix A, page 35. □

Note that it follows directly from the previous considerations that RR-equivalence is stronger than SR-equivalence, RMR-equivalence is stronger than SMR-equivalence and RE-equivalence is stronger than SE-equivalence. Figure 1 illustrates the strength comparison of all eight notions of program equivalence which are formally stated in Proposition 5 and in the following result:

Proposition 15. *The following holds:*

12

(1) $\equiv_{\mathsf{SM}} \prec \equiv_{\mathsf{SE}} \prec \equiv_{\mathsf{RE}} \prec \equiv_{\mathsf{RMR}} \prec \equiv_{\mathsf{RR}} \prec \equiv_{\mathsf{SU}}$ *and* $\models_{\mathsf{SE}} \prec \models_{\mathsf{RE}} \prec \models_{\mathsf{RMR}} \prec \models_{\mathsf{RR}} \prec \models_{\mathsf{SU}}$;

(2) $\equiv_{\mathsf{SE}} \prec \equiv_{\mathsf{SMR}} \prec \equiv_{\mathsf{SR}} \prec \equiv_{\mathsf{RR}}$ *and* $\models_{\mathsf{SE}} \prec \models_{\mathsf{SMR}} \prec \models_{\mathsf{SR}} \prec \models_{\mathsf{RR}}$;

(3) $\equiv_{\mathsf{SMR}} \prec \equiv_{\mathsf{RMR}}$ *and* $\models_{\mathsf{SMR}} \prec \models_{\mathsf{RMR}}$;

(4) $\equiv_{\mathsf{RE}} \not\preceq \equiv_{\mathsf{SMR}}$ *and* $\equiv_{\mathsf{SMR}} \not\preceq \equiv_{\mathsf{RE}}$ *and* $\models_{\mathsf{RE}} \not\preceq \models_{\mathsf{SMR}}$ *and* $\models_{\mathsf{SMR}} \not\preceq \models_{\mathsf{RE}}$;

(5) $\equiv_{\mathsf{RE}} \not\preceq \equiv_{\mathsf{SR}}$ *and* $\equiv_{\mathsf{SR}} \not\preceq \equiv_{\mathsf{RE}}$ *and* $\models_{\mathsf{RE}} \not\preceq \models_{\mathsf{SR}}$ *and* $\models_{\mathsf{SR}} \not\preceq \models_{\mathsf{RE}}$;

(6) $\equiv_{\mathsf{RMR}} \not\preceq \equiv_{\mathsf{SR}}$ *and* $\equiv_{\mathsf{SR}} \not\preceq \equiv_{\mathsf{RMR}}$ *and* $\models_{\mathsf{RMR}} \not\preceq \models_{\mathsf{SR}}$ *and* $\models_{\mathsf{SR}} \not\preceq \models_{\mathsf{RMR}}$.

Proof. See Appendix A, page 36.   □

4. Exception-Based Rule Update Operators

In this section we propose a generic scheme for specifying semantic rule update operators. We define instances of the scheme and show that they enjoy a number of plausible properties, ranging from the respect for support and fact update to syntax-independence and other semantic properties.

As suggested already, a program is semantically characterised by the set of sets of RE-models of its rules. Our update framework is based on a simple yet novel idea of introducing additional interpretations – *exceptions* – to the sets of RE-models of rules in the original program. The formalisation of this idea is straightforward: an exception-based update operator is characterised by an *exception function* ε that takes three inputs: the set of RE-models $[\![\pi]\!]_{\mathsf{RE}}$ of a rule $\pi \in \mathcal{P}$ and the semantic characterisations $\langle\!\langle\mathcal{P}\rangle\!\rangle_{\mathsf{RE}}$, $\langle\!\langle\mathcal{U}\rangle\!\rangle_{\mathsf{RE}}$ of the original and updating programs. Then it returns the interpretations that are to be introduced as exceptions to π, so the characterisation of the updated program contains the augmented set of RE-models

$$[\![\pi]\!]_{\mathsf{RE}} \cup \varepsilon\left([\![\pi]\!]_{\mathsf{RE}}, \langle\!\langle\mathcal{P}\rangle\!\rangle_{\mathsf{RE}}, \langle\!\langle\mathcal{U}\rangle\!\rangle_{\mathsf{RE}}\right) \quad . \tag{4}$$

Hence, the semantic characterisation of $\mathcal{P}$ updated by $\mathcal{U}$ is

$$\left\{\, [\![\pi]\!]_{\mathsf{RE}} \cup \varepsilon\left([\![\pi]\!]_{\mathsf{RE}}, \langle\!\langle\mathcal{P}\rangle\!\rangle_{\mathsf{RE}}, \langle\!\langle\mathcal{U}\rangle\!\rangle_{\mathsf{RE}}\right) \mid \pi \in \mathcal{P} \,\right\} \cup \langle\!\langle\mathcal{U}\rangle\!\rangle_{\mathsf{RE}} \quad . \tag{5}$$

In other words, rules from $\mathcal{P}$ are augmented with the respective exceptions and rules in $\mathcal{U}$ remain untouched.

From the syntactic viewpoint, we want a rule update operator $\oplus$ to return a program $\mathcal{P} \oplus \mathcal{U}$ with the semantic characterisation (5). This brings us to the following issue: What if no rule exists whose set of RE-models is equal to (4)? In that case, no rule corresponds to the augmented set of RE-models of a rule $\pi \in \mathcal{P}$, so the program $\mathcal{P} \oplus \mathcal{U}$ cannot be constructed. Moreover, such situations may occur quite frequently since a single rule has very limited expressivity. For instance, updating the fact $(p.)$ by the rule $(\sim p \leftarrow q, r.)$ may easily result in a set of RE-models expressible by the program $\{\, p \leftarrow \sim q., p \leftarrow \sim r. \,\}$ but not expressible by any single rule. To keep a firm link to operations on syntactic objects, we henceforth deal with this problem by allowing the inputs and output of rule update operators to be sets of rules *and programs*, which we dub *rule bases*.[8] In other words, the result of updating a rule, i.e. introducing exceptions to it, may be a set of rules, so the result of updating a program may be a rule base. Technically, a rule base can capture any possible result of an exception-based update due to Proposition 11.

Formally, a *rule base* is any set of rules and programs. Given a rule base $\mathcal{R}$, an interpretation J is a *model of* $\mathcal{R}$, denoted by $J \models \mathcal{R}$, if $J \models \Pi$ for all $\Pi \in \mathcal{R}$; $\mathcal{R}^J = \left\{\, \Pi^J \mid \Pi \in \mathcal{R} \,\right\}$; the set of *stable models of* $\mathcal{R}$

---

[8] We allow for individual rules in a rule base out of convenience only. A single rule π in a rule base $\mathcal{R}$ is treated exactly the same way as if $\mathcal{R}$ contained the singleton program $\{\, \pi \,\}$.

consists of all interpretations J such that J is a subset-minimal model of $\mathcal{R}^J$; $\langle\!\langle\mathcal{R}\rangle\!\rangle_{\mathsf{SE}} = \{\ [\![\Pi]\!]_{\mathsf{SE}} \mid \Pi \in \mathcal{R}\ \}$ and $[\![\mathcal{R}]\!]_{\mathsf{SE}} = \bigcap\langle\!\langle\mathcal{R}\rangle\!\rangle_{\mathsf{SE}}$; $\langle\!\langle\mathcal{R}\rangle\!\rangle_{\mathsf{RE}} = \{\ [\![\Pi]\!]_{\mathsf{RE}} \mid \Pi \in \mathcal{R}\ \}$ and $[\![\mathcal{R}]\!]_{\mathsf{RE}} = \bigcap\langle\!\langle\mathcal{R}\rangle\!\rangle_{\mathsf{RE}}$. All notions of program equivalence and entailment are naturally extended to rule bases and any rule update operator $\oplus$ is generalised to sequences of rule bases $\langle\mathcal{R}_i\rangle_{i<n}$ as follows: $\bigoplus\langle\rangle = \emptyset$ and $\bigoplus\langle\mathcal{R}_i\rangle_{i<n+1} = (\bigoplus\langle\mathcal{R}_i\rangle_{i<n}) \oplus \mathcal{R}_n$.

Note that a program is a special case of a rule base. Each element Π of a rule base, be it a rule or a program, represents an *atomic* piece of information and exception-based update operators view and manipulate Π only through its set of RE-models $[\![\Pi]\!]_{\mathsf{RE}}$. Due to this, we refer to all such elements Π as *rules*, even if formally they may actually be programs.

Having resolved this issue, we can proceed to the definition of an exception-based rule update operator.

Definition 16 (Exception-Based Rule Update Operator). Given an exception function ε, a rule update operator $\oplus$ is ε-*based* if for all rule bases $\mathcal{R}, \mathcal{U}$, $\langle\!\langle\mathcal{R} \oplus \mathcal{U}\rangle\!\rangle_{\mathsf{RE}}$ is equal to (5). Also, $\oplus$ is *exception-based* if it is ε-based for some exception function ε.

Note that for each exception function ε there is a whole class of ε-based rule update operators that differ in the syntactic representations of the sets of RE-models in (5). For instance, when working over the set of propositional symbols $\mathcal{A} = \{\,p, q\,\}$ and considering some exception-based operator $\oplus$, the exception function may specify that for some programs $\mathcal{P}, \mathcal{U}$, the program $\mathcal{P} \oplus \mathcal{U}$ contains a rule or program representing the set of RE-models $\mathcal{M} = \{\ (\emptyset, \emptyset), (\emptyset, p), (p, p), (\emptyset, q), (\emptyset, pq), (p, pq), (pq, pq)\ \}$.[9] This set can be represented by the rule $\pi = (p \leftarrow q.)$ or, alternatively, by the rule $\sigma = (p; \sim q \leftarrow q.)$, or even by the program $\mathcal{Q} = \{\ (p \leftarrow q.), (p \leftarrow p, \sim q.)\ \}$, and the exception function does not specify which syntactic representation of the set should be used in $\mathcal{P} \oplus \mathcal{U}$.

4.1. Simple Exception Functions and Their Syntactic Properties

General exception functions, as introduced above, have as inputs the entire semantic characterisations of the original as well as updating programs ($\langle\!\langle\mathcal{P}\rangle\!\rangle_{\mathsf{RE}}$ and $\langle\!\langle\mathcal{U}\rangle\!\rangle_{\mathsf{RE}}$, respectively) when determining exceptions to any single rule. As it turns out, all this information is not strictly necessary in order to capture rule update operators enjoying a range of plausible syntactic as well as semantic properties.

Thus, further study of general exception functions is left for future research and in this paper we concentrate on a simpler, constrained class of exception functions that is nevertheless powerful enough to serve as a basis for well-behaved semantic rule update operators. Not only does this lead to simpler definitions, but the study of restricted classes of exception functions is essential in order to understand their expressivity, i.e. the types of update operators they are able to capture.

More particularly, we study exception functions that produce exceptions based on conflicts between pairs of rules, one from the original and one from the updating program, while ignoring the context in which these rules are situated. Formally, an exception function ε is *simple* if for all $\mathcal{M} \subseteq \mathcal{X}$ and $\mathcal{S}, \mathcal{T} \subseteq 2^{\mathcal{X}}$,

$$\varepsilon(\mathcal{M}, \mathcal{S}, \mathcal{T}) = \bigcup\nolimits_{\mathcal{N} \in \mathcal{T}} \delta(\mathcal{M}, \mathcal{N})$$

where $\delta : 2^{\mathcal{X}} \times 2^{\mathcal{X}} \to 2^{\mathcal{X}}$ is a *local exception function*. If $\oplus$ is an ε-based rule update operator, then we also say that $\oplus$ is δ-*based* and that $\oplus$ is *simple*.

As we shall see, in spite of their local nature, particular simple exception functions generate rule update operators that satisfy the syntactic properties laid out in Section 2 and are closely related to the JU- and UA-semantics for DLPs.

---

[9]Sometimes we omit the usual set notation when we write interpretations, e.g. instead of $\{\,p, q\,\}$ we may simply write pq.

The inspiration for defining concrete local exception functions δ comes from rule update semantics based on causal rejection. But since the relevant concepts, such as that of a *conflict* or *rule rejection*, rely on rule syntax to which an exception function has no direct access, our first objective is to find similar concepts on the semantic level. In particular, we need to define conditions under which two sets of RE-models are in conflict. But first we introduce two preparatory concepts.

We define a *truth value substitution* as follows: Given an interpretation J, an atom p and a truth value $\mathsf{V} \in \{\mathsf{T}, \mathsf{U}, \mathsf{F}\}$, by $J[\mathsf{V}/p]$ we denote the three-valued interpretation X such that $X(p) = \mathsf{V}$ and $X(q) = J(q)$ for all atoms $q \neq p$.

This enables us to introduce the main concept needed for defining a conflict between two sets of three-valued interpretations. Given a set of three-valued interpretations $\mathcal{M}$, an atom p, a truth value V_0 and a two-valued interpretation J, we say that $\mathcal{M}$ *forces p to have the truth value* V_0 *w.r.t.* J, denoted by $\mathcal{M}^J(p) = \mathsf{V}_0$, if

$$J[\mathsf{V}/p] \in \mathcal{M} \text{ if and only if } \mathsf{V} = \mathsf{V}_0 \ .$$

In other words, the three-valued interpretation $J[\mathsf{V}_0/p]$ must be the unique member of $\mathcal{M}$ that either coincides with J or differs from it only in the truth value of p. Note that $\mathcal{M}^J(p)$ stays undefined in case no V_0 with the above property exists.

Two sets of three-valued interpretations $\mathcal{M}, \mathcal{N}$ are *in conflict on atom p w.r.t.* J, denoted by $\mathcal{M} \bowtie_p^J \mathcal{N}$, if both $\mathcal{M}^J(p)$ and $\mathcal{N}^J(p)$ are defined and $\mathcal{M}^J(p) \neq \mathcal{N}^J(p)$. The following example illustrates all these concepts.

Example 17. Consider rules $\pi = (p.)$, $\sigma = (\sim p \leftarrow \sim q.)$ with the respective sets of RE-models

$$\mathcal{M} = \{\, (p,p), (p,pq), (pq,pq) \,\} \quad \text{and} \quad \mathcal{N} = \{\, (\emptyset,\emptyset), (\emptyset,q), (q,q), (\emptyset,pq), (p,pq), (q,pq), (pq,pq) \,\} \ .$$

Intuitively, $\mathcal{M}$ forces p to T w.r.t. all interpretations and $\mathcal{N}$ forces p to F w.r.t. interpretations in which q is false. Formally it follows that $\mathcal{M}^\emptyset(p) = \mathsf{T}$ because (p,p) belongs to $\mathcal{M}$ and neither $(\emptyset,p)$ nor $(\emptyset,\emptyset)$ belongs to $\mathcal{M}$. Similarly, it follows that $\mathcal{N}^\emptyset(p) = \mathsf{F}$. Hence $\mathcal{M} \bowtie_p^\emptyset \mathcal{N}$. Using similar arguments we can conclude that $\mathcal{M} \bowtie_p^p \mathcal{N}$. However, it does not hold that $\mathcal{M} \bowtie_p^{pq} \mathcal{N}$ because $\mathcal{N}^{pq}(p)$ is undefined.

In order to define a particular local exception function based on causal rejection, it only remains to decide which three-valued interpretations become exceptions when a conflict w.r.t. an interpretation J occurs. One intuition we can draw on is the relationship between RE-models and stable models (c.f. Proposition 10): J is a stable model of a program if and only if (J,J) is the unique RE-model of the program whose second component is J. So given an original rule π with $\mathcal{M} = [\![\pi]\!]_{\mathsf{RE}}$ and an updating rule σ with $\mathcal{N} = [\![\sigma]\!]_{\mathsf{RE}}$ where a conflict occurs between $\mathcal{M}$ and $\mathcal{N}$ w.r.t. J, π needs to be weakened so that it cannot influence whether J becomes a stable model of the updated program or not. In other words, we need to introduce all three-valued interpretations whose second component is J as exceptions to that rule. Formally:

Definition 18 (Exception Function δ_{a}). The local exception function δ_{a} is for all $\mathcal{M}, \mathcal{N} \subseteq \mathfrak{X}$ defined by:

$$\delta_{\mathsf{a}}(\mathcal{M}, \mathcal{N}) = \{\, (I,J) \in \mathfrak{X} \mid \exists p : \mathcal{M} \bowtie_p^J \mathcal{N} \,\} \ .$$

Theorem 19 (Syntactic Properties of δ_{a}). *Every δ_{a}-based rule update operator respects support and fact update and it also respects causal rejection and acyclic justified update w.r.t. DLPs of length at most two.*

Proof. See Appendix B, page 43. □

This means that δ_a-based rule update operators enjoy a combination of desirable syntactic properties that operators based on SE-models cannot [69]. However, these operators diverge from causal rejection, even on acyclic DLPs, when more than one update is performed.

Example 20. Consider again the rules π, σ and their sets of RE-models $\mathcal{M}$, $\mathcal{N}$ from Example 17 and some δ_a-based rule update operator $\oplus$. Then $\langle\!\langle \{ \pi \} \oplus \{ \sigma \} \rangle\!\rangle_{\mathsf{RE}}$ will contain two elements, $\mathcal{M}'$ and $\mathcal{N}$, where $\mathcal{M}' = \mathcal{M} \cup \delta_a(\mathcal{M}, \mathcal{N}) = \mathcal{M} \cup \{ (\emptyset, \emptyset), (\emptyset, p), (p, p) \}$. An additional update by the fact $\{ q. \}$ then leads to the characterisation $\langle\!\langle \bigoplus \langle \{ \pi \}, \{ \sigma \}, \{ q. \} \rangle \rangle\!\rangle_{\mathsf{RE}}$ which contains three elements: $\mathcal{M}''$, $\mathcal{N}$ and $[\![q.]\!]_{\mathsf{RE}}$ where $\mathcal{M}'' = \mathcal{M}' \cup \{ (\emptyset, q), (q, q) \}$. Furthermore, due to Proposition 10, $J = \{ q \}$ is a stable model of the program $\bigoplus \langle \{ \pi \}, \{ \sigma \}, \{ q. \} \rangle$ because (q, q) belongs to $[\![\bigoplus \langle \{ \pi \}, \{ \sigma \}, \{ q. \} \rangle]\!]_{\mathsf{RE}}$ and $(\emptyset, q)$ does not. However, J does not respect causal rejection and it is neither a JU- nor a UA-model of $\langle \{ \pi \}, \{ \sigma \}, \{ q. \} \rangle$.

This shortcoming of δ_a largely stems from the asymmetry of its definition – when a conflict occurs w.r.t. J, atoms that are true in J may become undefined in an exception (I, J) but atoms that are false must remain false. Eliminating this asymmetry, by allowing for exceptions (I, K) with $I \subseteq J \subseteq K$, and choosing the introduced exceptions more carefully, leads to more well-behaved exception functions, defined as follows:

Definition 21 (Exception Functions δ_b, δ_c, δ_d and δ_e)**.** The local exception functions δ_b, δ_c, δ_d and δ_e are for all $\mathcal{M}, \mathcal{N} \subseteq \mathfrak{X}$ defined by:

$$\delta_b(\mathcal{M}, \mathcal{N}) = \left\{ (I, K) \in \mathfrak{X} \mid \exists J \exists p : \mathcal{M} \bowtie_p^J \mathcal{N} \wedge I \subseteq J \subseteq K \wedge (p \in K \setminus I \Rightarrow K = J) \right\} \ ,$$

$$\delta_c(\mathcal{M}, \mathcal{N}) = \left\{ (I, K) \in \mathfrak{X} \mid \exists J \exists p : \mathcal{M} \bowtie_p^J \mathcal{N} \wedge I \subseteq J \subseteq K \wedge (p \in K \setminus I \Rightarrow K = J \wedge (J, J) \notin \mathcal{M}) \right\} \ ,$$

$$\delta_d(\mathcal{M}, \mathcal{N}) = \begin{cases} \mathfrak{X} & \text{if } \mathcal{M} = \mathcal{N} \text{ ;} \\ \delta_b(\mathcal{M}, \mathcal{N}) & \text{otherwise ,} \end{cases} \qquad \delta_e(\mathcal{M}, \mathcal{N}) = \begin{cases} \mathfrak{X} & \text{if } \mathcal{M} = \mathcal{N} \text{ ;} \\ \delta_c(\mathcal{M}, \mathcal{N}) & \text{otherwise .} \end{cases}$$

The functions δ_b and δ_d introduce more exceptions than δ_a while δ_c and δ_e eliminate some of those returned by δ_a and add some additional ones. The difference between δ_b and δ_d, and similarly also between δ_c and δ_e, is in that δ_d additionally "wipes out" rules from the original program that are repeated in the update by introducing all interpretations as exceptions to them, rendering them tautological. This will turn out to be significant later when we examine semantic properties of simple exception-based rule update operators. In all four functions, a conflict on p w.r.t. J leads to the introduction of interpretations in which atoms either maintain the truth value they had in J, or they become undefined. Additionally, extra conditions are imposed on the case when p becomes undefined. Interestingly, this leads to operators that satisfy all syntactic properties.

Theorem 22 (Syntactic Properties of δ_b, δ_c, δ_d and δ_e)**.** *Let $\oplus$ be a δ_b-, δ_c-, δ_d- or δ_e-based rule update operator. Then $\oplus$ respects support, fact update, causal rejection and acyclic justified update.*

Proof. See Appendix B, page 47. □

Furthermore, it is worth noting that even on programs with cycles, δ_b- and δ_d-based operators are very closely related to the JU-semantics and, similarly, δ_c- and δ_e-based operators to the UA-semantics. They diverge from the syntax-based semantics only on rules with an appearance of the same atom in both the head and body. Formally, we say that a rule is a *local cycle* if $(\mathsf{H}_\pi^+ \cup \mathsf{H}_\pi^-) \cap (\mathsf{B}_\pi^+ \cup \mathsf{B}_\pi^-) \neq \emptyset$.

Theorem 23 (Characterisation of JU- and UA-Semantics Using Exception Functions)**.** *Let P be a DLP, J an interpretation, $\oplus_1$ a δ_b- or δ_d-based rule update operator and $\oplus_2$ a δ_c- or δ_e-based rule update operator. Then,*

- $[\![\bigoplus_1 \boldsymbol{P}]\!]_{\mathsf{SM}} \subseteq [\![\boldsymbol{P}]\!]_{\mathsf{JU}}$ *and* $[\![\bigoplus_2 \boldsymbol{P}]\!]_{\mathsf{SM}} \subseteq [\![\boldsymbol{P}]\!]_{\mathsf{UA}}$;

- *if* $\mathsf{all}(\boldsymbol{P})$ *contains no local cycles, then* $[\![\boldsymbol{P}]\!]_{\mathsf{JU}} \subseteq [\![\bigoplus_1 \boldsymbol{P}]\!]_{\mathsf{SM}}$ *and* $[\![\boldsymbol{P}]\!]_{\mathsf{UA}} \subseteq [\![\bigoplus_2 \boldsymbol{P}]\!]_{\mathsf{SM}}$.

Proof. See Appendix B, page 47. ☐

This means that up to the marginal case of local cycles, δ_{b} and δ_{d} can be seen as semantic characterisations of the JU-semantics and, similarly, δ_{c} and δ_{e} characterise the UA-semantics. The exceptional cases when *less* stable models are found than using the traditional semantics occur when the DLP contains tautological rules or rules with the negation of their head in the body. The former can be seen as a strict improvement as it introduces immunity to tautologies. The latter is an unavoidable consequence of the fact that exception-based operators only manipulate RE-models which are unable to distinguish between a constraint ($\leftarrow p.$) and a rule ($\sim p \leftarrow p.$) The traditional approaches do distinguish between them in that the former rule can never be rejected while the latter can.

This tight relationship also sheds new light on the problem of *state condensing*, discussed in more detail in Section 5.

4.2. Semantic Properties

We proceed by examining further properties of rule update operators – of those based on simple exception functions in general, and of the δ_{a}-, δ_{b}-, δ_{c}-, δ_{d}- and δ_{e}-based ones in particular. The properties we consider in this section are *semantic* in that they put conditions on the *models* of a result of an update and do not need to refer to the syntax of the original and updating programs. Our results are summarised in Table 2 and in the following we explain and discuss them. The interested reader may find all the proofs in Appendix B.5 starting on page 48.

Traditional Semantic Properties of Rule Updates

The properties in the upper part of Table 2 were introduced in [5, 20, 24]. We formalise all of them for rule bases $\mathcal{R}, \mathcal{S}, \mathcal{U}, \mathcal{V}$ and a rule update operator $\oplus$ and each can actually be seen as a *meta-property* that is instantiated once we adopt a particular notion of program equivalence. Therefore, each row of Table 2 has eight cells that stand for particular instantiations of the property. This provides a more complete picture of how simple rule update operators, properties and program equivalence are interrelated.

Unless stated otherwise (in a footnote), each tick (✓) signifies that the property in question holds for *all* simple rule update operators. A missing tick signifies that the property does not generally hold for simple rule update operators, and in particular there are δ_{a}-, δ_{b}-, δ_{c}-, δ_{d}- and δ_{e}-based operators for which it is violated. A tick is smaller if it is a direct consequence of a preceding larger tick in the same row and of the interrelations between the program equivalence and entailment relations (c.f. Figure 1).

At a first glance, it is obvious that none of the semantic properties is satisfied under SU-equivalence. This is because the conditions placed on a rule update operator by an exception function are at the semantic level, while SU-equivalence effectively compares programs syntactically. For instance, an exception-based operator $\oplus$, for any exception function ε, may behave as follows: $\emptyset \oplus \{ \sim p \leftarrow p. \} = \{ \leftarrow p. \}$. This is because the rules before and after update are RE-equivalent. However, due to the fact that the programs $\{ \sim p \leftarrow p. \}$ and $\{ \leftarrow p. \}$ are considered different under SU-equivalence, $\oplus$ cannot satisfy (Initialisation) w.r.t. SU-equivalence. The situation with all other properties is analogous.

As for the other notions of equivalence, we separately discuss each group of related properties:

(Initialisation) and **(Disjointness)**: These properties are satisfied "by construction", regardless of which simple rule update operator we consider and of which notion of equivalence we pick.

Table 2: Semantic properties of simple rule update operators

Property	Formalisation		Type of $\equiv$, $\models$ and $\llbracket \cdot \rrbracket$						
		SU	RR	SR	RMR	SMR	RE	SE	SM
(Initialisation)	$\emptyset \oplus \mathcal{U} \equiv \mathcal{U}$.		**✓**	✓	✓	✓	✓	✓	✓
(Disjointness)	If $\mathcal{R}, \mathcal{S}$ are over disjoint alphabets, then $(\mathcal{R} \cup \mathcal{S}) \oplus \mathcal{U} \equiv (\mathcal{R} \oplus \mathcal{U}) \cup (\mathcal{S} \oplus \mathcal{U})$.		**✓**	✓	✓	✓	✓	✓	✓
(Non-interference)[†]	If $\mathcal{U}, \mathcal{V}$ are over disjoint alphabets, then $(\mathcal{R} \oplus \mathcal{U}) \oplus \mathcal{V} \equiv (\mathcal{R} \oplus \mathcal{V}) \oplus \mathcal{U}$.		**✓**[abcde]	✓[abcde]	✓[abcde]	✓[abcde]	✓[abcde]	✓[abcde]	✓[abcde]
(Tautology)	If $\mathcal{U}$ is tautological, then $\mathcal{R} \oplus \mathcal{U} \equiv \mathcal{R}$.		**✓**[⋆]	✓[⋆]	✓[⋆]	✓[⋆]	✓[⋆]	✓[⋆]	✓[⋆]
(Immunity to Tautologies)	If $\mathcal{S}$ and $\mathcal{V}$ are tautological, then $(\mathcal{R} \cup \mathcal{S}) \oplus (\mathcal{U} \cup \mathcal{V}) \equiv \mathcal{R} \oplus \mathcal{U}$.		**✓**[⋆]	✓[⋆]	✓[⋆]	✓[⋆]	✓[⋆]	✓[⋆]	✓[⋆]
(Idempotence)	$\mathcal{R} \oplus \mathcal{R} \equiv \mathcal{R}$.		✓[de]	✓[de]	**✓**	✓	✓	✓	✓
(Absorption)	$(\mathcal{R} \oplus \mathcal{U}) \oplus \mathcal{U} \equiv \mathcal{R} \oplus \mathcal{U}$.		✓[de]	✓[de]	**✓**[bcde]	✓[bcde]	✓[bcde]	✓[bcde]	✓[bcde]
(Augmentation)[†]	If $\mathcal{U} \subseteq \mathcal{V}$, then $(\mathcal{R} \oplus \mathcal{U}) \oplus \mathcal{V} \equiv \mathcal{R} \oplus \mathcal{V}$.		✓[de]	✓[de]	**✓**[bcde]	✓[bcde]	✓[bcde]	✓[bcde]	✓[bcde]
(Associativity)	$\mathcal{R} \oplus (\mathcal{U} \oplus \mathcal{V}) \equiv (\mathcal{R} \oplus \mathcal{U}) \oplus \mathcal{V}$.								
(P1)	$\mathcal{R} \oplus \mathcal{U} \models \mathcal{U}$		**✓**	✓	✓	✓	✓	✓	n/a
(P2.$\top$)	$\mathcal{R} \oplus \emptyset \equiv \mathcal{R}$		**✓**	✓	✓	✓	✓	✓	✓
(P2.1)	$\mathcal{R} \cup \mathcal{U} \models \mathcal{R} \oplus \mathcal{U}$					**✓**	✓	✓	n/a
(P2.2)	$(\mathcal{R} \cup \mathcal{U}) \oplus \mathcal{U} \models \mathcal{R}$								n/a
(P3)	If $\llbracket \mathcal{R} \rrbracket \neq \emptyset$ and $\llbracket \mathcal{U} \rrbracket \neq \emptyset$, then $\llbracket \mathcal{R} \oplus \mathcal{U} \rrbracket \neq \emptyset$	n/a	n/a	n/a	n/a	n/a			
(P4)	If $\mathcal{R} \equiv \mathcal{S}$ and $\mathcal{U} \equiv \mathcal{V}$, then $\mathcal{R} \oplus \mathcal{U} \equiv \mathcal{S} \oplus \mathcal{V}$		**✓**[⋆]						
(P5)	$(\mathcal{R} \oplus \mathcal{U}) \cup \mathcal{V} \models \mathcal{R} \oplus (\mathcal{U} \cup \mathcal{V})$					**✓**	✓	✓	n/a
(P6)	If $\mathcal{R} \oplus \mathcal{U} \models \mathcal{V}$ and $\mathcal{R} \oplus \mathcal{V} \models \mathcal{U}$, then $\mathcal{R} \oplus \mathcal{U} \equiv \mathcal{R} \oplus \mathcal{V}$								n/a

[a] Holds if $\oplus$ is δ_a-based.

[b] Holds if $\oplus$ is δ_b-based.

[c] Holds if $\oplus$ is δ_c-based.

[d] Holds if $\oplus$ is δ_d-based.

[e] Holds if $\oplus$ is δ_e-based.

[⋆] Holds if $\oplus$ is based on a local exception function δ such that $\delta(\mathcal{M}, \mathcal{X}) \subseteq \mathcal{M}$ for all $\mathcal{M} \subseteq \mathcal{X}$. This is satisfied by δ_a, δ_b, δ_c, δ_d and δ_e.

[†] Results on this line hold only if $\mathcal{R}, \mathcal{U}, \mathcal{V}$ are non-disjunctive programs.

(Tautology) and (Immunity to Tautologies): These are naturally satisfied by all simple update operators that do not introduce exceptions merely due to the presence of a tautological rule in the updating program. In particular, both properties are satisfied by δ_a-, δ_b-, δ_c-, δ_d- and δ_e-based operators. Note that these properties are generally acknowledged as very desirable although most existing rule update semantics fail to comply with them [3, 20, 24, 48, 55, 60, 75].

(Non-interference): This property is satisfied by δ_a-, δ_b-, δ_c-, δ_d- and δ_e-based rule update operators. However, this is only the case when non-disjunctive programs are considered, pointing towards one of the important open problems faced by state-of-the-art research on rule updates: examples, desirable properties and methods for updating *disjunctive* programs. Insights in this direction should shed light on whether (Non-interference) is desirable in the disjunctive case.

(Idempotence), (Absorption) and (Augmentation): These are the only properties that reveal differences amongst the different exception functions. They are not satisfied by δ_a-, δ_b- and δ_c-based operators under SR- and RR-equivalence because a program updated by its subset may contain weakened versions of the original rules. Since such rules are not part of the original program, the programs before and after update are considered to be different under SR- and RR-equivalence. This problem is dodged in δ_d and δ_e by completely obliterating original rules that appear in the update.

This also indicates that SR- and RR-equivalence are slightly *too strong* for characterising updates because programs such as $\{\,p.\,\}$ and $\{\,p., p \leftarrow q.\,\}$ are not considered equivalent even though we expect the same behaviour from them when they are updated. We speculated in Section 3 that it might be possible to address this issue by adopting the weaker SMR- or RMR-equivalence. However, it turns out that due to the monotonic nature of SE- and RE-models, these equivalence relations are *too weak*: programs such as $\{\,p.\,\}$ and $\{\,p., q \leftarrow {\sim} p.\,\}$ are SMR- and RMR-equivalent but, when updated by $\{\,{\sim} p.\,\}$, any rule update operator respecting fact update and causal rejection must provide the stable model $\emptyset$ in the former case, which violates causal rejection in the latter case.

Moreover, δ_a-based operators fail to satisfy (Absorption) and (Augmentation). Along with Example 20, this indicates that δ_a does not correctly handle iterated updates.

(Associativity): This is one of the few properties that is not satisfied by any of the defined classes of operators. It is closely related to the question of whether *rejected rules are allowed to reject*. (Associativity) can be seen as postulating that an update operator must behave the same way regardless of whether rejected rules are allowed to reject or not. As witnessed by the JU- and UA-semantics (c.f. equation 2), rule update semantics tend to generate unwanted models when rejected rules are not allowed to reject.

Belief Update Postulates Reformulated

The lower part of Table 2 contains a reformulation of the belief update postulates (B1) – (B6) for rule bases. We omit the last two postulates as they require program disjunction and it is not clear how to obtain it appropriately. Note also that (B7) has been heavily criticised in the literature as being mainly a means to achieve formal results instead of an intuitive principle [34] and though (B8) reflects the basic intuition behind belief update – that of updating each model independently of the others – such a point of view is hardly transferable to knowledge represented using rules because a single model, be it a classical, stable, SE- or RE-model, fails to encode the interdependencies between literals expressed in rules that are necessary for properties such as support.

Since we did not define SM-entailment, postulates that refer to it have the SM column marked as "n/a". Now we turn to the individual postulates.

(P1) and **(P2.⊤)**: Similarly as (Initialisation) and (Disjointness), these postulates are satisfied by any simple rule update operator and under all notions of equivalence.

(P2.1) and **(P5)**: Postulate (P2.1) is not satisfied under SR- and RR-equivalence for the same reasons, described above, that prevent δ_a- and δ_b-based operators from satisfying (Idempotence). The situation with (P5) is the same since it implies (P2.1) in the presence of (P2.⊤) [34].

(P2.2) and **(P6)**: Postulate (P2.2) requires that $\{\,p., \sim p.\,\} \oplus \{\,\sim p.\,\} \models p$ which, in the presence of (P1), amounts to postulating that one can never recover from an inconsistent state, contrary to most existing rule update semantics which do allow for recovery from such states. The case of (P6) is the same since it implies (P2.2) in the presence of (P1) and (P2.⊤) [34].

(P3): This postulate relies on a function that returns the set of models of a rule base. Thus, $[\![\cdot]\!]_{\mathsf{SM}}$, $[\![\cdot]\!]_{\mathsf{SE}}$ and $[\![\cdot]\!]_{\mathsf{RE}}$ can be used for this purpose and the other columns in the corresponding row in Table 2 make little sense, so they are marked as "n/a". Furthermore, this postulate is not satisfied by any of the defined classes of exception-based operators. It is also one of the principles that most existing approaches to rule update chronically fail to satisfy. In order to satisfy it, a context-aware exception function would have to be used because conflicts may arise in a set of more than two rules that are pairwise consistent. For instance, when updating $\{\,p.\,\}$ by $\{\,q \leftarrow p., \sim q \leftarrow p.\,\}$, one would somehow need to detect and resolve the joint conflict between these three rules. This is however impossible with a simple exception function because it only considers conflicts between pairs of rules, one from the original program and one from the update.

(P4): This postulate requires update operators to be syntax-independent. In this context it is useful to consider the following weaker principles:

> **(P4.1)** If $\mathcal{P} \equiv \mathcal{Q}$, then $\mathcal{P} \diamond \mathcal{U} \equiv \mathcal{Q} \diamond \mathcal{U}$.   **(P4.2)** If $\mathcal{U} \equiv \mathcal{V}$, then $\mathcal{P} \diamond \mathcal{U} \equiv \mathcal{P} \diamond \mathcal{V}$.

The failure to satisfy (P4.1) under SM-, SE-, RE-, SMR- and RMR-equivalence is inevitable if properties such as support, fact update and causal rejection are to be respected (c.f. [69] and the above discussion of (Idempotence), (Absorption) and (Augmentation)). Furthermore, (P4.1) is also violated under SR-equivalence due to the fact that a constraint such as $(\leftarrow p.)$ cannot be weakened by the introduced exception functions while the fact $(\sim p.)$ can, although it is strongly equivalent to the constraint.

Similar arguments can be used to show that the principle (P4.2) is not satisfied under SM-, SE-, RE, SMR- and RMR-equivalence. We only need to observe that any δ_a-, δ_b-, δ_c-, δ_d- or δ_e-based operator $\oplus$ satisfies $\emptyset \oplus \mathcal{P} \equiv_{\mathsf{RR}} \mathcal{P}$, so each example used to show that (P4.1) is not desirable, involving two updates $\mathcal{P}_1 \oplus \mathcal{U}$ and $\mathcal{P}_2 \oplus \mathcal{U}$, can be reused to show the same for (P4.2) by instead considering the updates $\emptyset \oplus \mathcal{P}_1 \oplus \mathcal{U}$ and $\emptyset \oplus \mathcal{P}_1 \oplus \mathcal{U}$. Additionally, (P4.2) is not satisfied under SR-equivalence due to the fact that updates such as $\{\,\sim p \leftarrow q.\,\}$, $\{\,\sim q \leftarrow p.\,\}$ and $\{\,\leftarrow p, q.\,\}$ have different effects on the program $\{\,p., q.\,\}$.

These observations indicate that (P4.1) and (P4.2), and thus also (P4), are too strong under SM-, SE-, RE-, SMR- and RMR-equivalence. Under SR-equivalence, they are incompatible with operators that solve conflicts based on the heads of rules. On the other hand, due to the semantic underpinning of simple rule update operators, (P4) is satisfied by most of them, including all δ_a-, δ_b-, δ_c-, δ_d- and δ_e-based ones, under RR-equivalence.

Summary

The results in Table 2 indicate that simple rule update operators satisfy a number of essential semantic properties by design. This is especially important for properties such as (Tautology), (Immunity to Tautolo-

gies) and (P2.⊤), generally acknowledged as very desirable although most existing rule update semantics fail to comply with them [3, 20, 24, 48, 55, 60, 75].

Some of the semantic properties, namely (Idempotence), (Absorption) and (Augmentation), outline the main differences between the particular local exception functions that we have defined. Although δ_a is the function with the simplest definition, it does not satisfy any of these properties, indicating that it cannot deal well with iterated updates.

Our results also show that a number of KM postulates are not satisfied by the defined rule update operators. Coincidentally, with the exception of (B3), the violated postulates are the ones that are deemed controversial in the belief update community [10, 11, 22, 34].

Perhaps most importantly, Table 2 introduces a novel way of viewing the various semantic properties, where each property is parametrised by a particular notion of program equivalence and/or entailment. This richer view should prove useful in further research on rule update operators and their semantic properties.

5. State Condensing

Our results about semantic rule update operators bring along a new point of view on traditional approaches to rule updates. Particularly interesting is the fact that a semantic rule update operator, following the belief update tradition, is a binary function on the set of programs over the same alphabet. This guarantees that after an update, the original program is *replaced* by a new program which continues to be used in its place, and the process can be easily iterated if the need arises. Even though this perspective is very natural, even fundamental, it has been largely neglected in the ample body of work on rule updates. Existing semantics typically proceed by characterising the models of the update and, at most, either describe a set of answer-set programs that could represent the update, instead of only one, or produce an answer-set program written in a language extended with a considerable amount of new atoms, making it difficult to understand and to further update.

For instance, semantics based on causal rejection [3, 5, 24, 48, 55], such as the JU- and UA-semantics, assign models to sequences of non-disjunctive programs and sometimes admit non-minimal models which no non-disjunctive program can capture. Consequently, they must resort to the introduction of additional meta-level atoms in order to construct a single program whose stable models correspond to the models assigned to the sequence, leading to difficulties with iterating the update process. Producing a program from a more expressive class of logic programs would result in the inability to perform another update as well because the update semantics are not defined for such programs.

A different approach in [60] deals with program updates by borrowing ideas from literature on *belief revision* and utilising an abductive framework to accomplish such updates. In this case, multiple alternative programs can be the result of an update and no mechanism is provided to choose among them.

A somewhat similar situation occurs with the approach of [75] where intricate syntactic transformations are combined with a semantics for prioritised logic programs that ultimately leads to a set of logic programs. Since all of these programs together represent the result of the update, it is once again unclear how to construct a single program that combines all of them.

The rule update semantics suggested in [20] are also based on syntactic transformations into a logic program with preferences among rules, but in contrast with [75], the semantics of such programs is defined by directly specifying their preferred stable models and not by translation into an ordinary program (or a set thereof). Thus, although an actual syntactic object is constructed that represents the update, it needs to be interpreted in a richer formalism to take into account preferences among rules.

Finally, frameworks that specify program updates by manipulating dependencies on default assumptions induced by rules [46, 61, 62] are mainly concerned with identifying the effects of irrelevant updates and

other theoretical properties of the stable models assigned to a pair or sequence of programs. They do not consider representing the result of an update by a single program.

In this section, we unravel the true potential of specifying updates as binary operators on some class of programs. Despite the fact that existing program update semantics do not seem compatible with this point of view, we show that at least the foundational JU- and UA-semantics *can* be viewed in this manner.

Theorem 23 already provides part of the solution as it shows that any DLP can be *condensed* into a *single rule base* over the same alphabet that behaves just as the original DLP when further updates are performed on it. Formally, this can be stated as follows:

Corollary 24 (State Condensing into a Rule Base). *There exist rule update operators $\oplus_1$ and $\oplus_2$ such that for every DLP $P = \langle \mathcal{P}_i \rangle_{i<n}$ without local cycles and all $j < n$ there exist rule bases $\mathcal{R}_1$ and $\mathcal{R}_2$ with*

$$\llbracket P \rrbracket_{\mathsf{JU}} = \llbracket \bigoplus_1 \langle \mathcal{R}_1, \mathcal{P}_{j+1}, \ldots, \mathcal{P}_{n-1} \rangle \rrbracket_{\mathsf{SM}} \qquad and \qquad \llbracket P \rrbracket_{\mathsf{UA}} = \llbracket \bigoplus_2 \langle \mathcal{R}_2, \mathcal{P}_{j+1}, \ldots, \mathcal{P}_{n-1} \rangle \rrbracket_{\mathsf{SM}} \ .$$

Proof. See Appendix B, page 47. □

However, this result is not completely satisfactory since it relies on rule bases instead of standard classes of logic programs, does not provide explicit definitions of operators $\oplus_1$ and $\oplus_2$, and does not apply to DLPs with local cycles.

In the following, we address all these issues by defining specific rule update operators that faithfully characterise the JU- and UA-semantics – when applied to an arbitrary DLP P, they produce a program whose stable models coincide with JU- and UA-models of P, respectively. In this way, the new operators provide a way to condense any DLP into a single program that includes all relevant information about the DLP, not only to identify its stable models, but also for the purpose of performing further updates. Thereby, we solve the long-standing problem known as *state condensing* from the literature on causal rejection semantics for rule updates.

To achieve this, our operators must deal with a more general class of programs than non-disjunctive ones. First we define simple and elegant operators on subclasses of *nested logic programs* [71] with the required property. Subsequently, we show that the full expressivity of nested programs is not necessary for this purpose by defining an additional pair of operators that produce *disjunctive logic programs* and still maintain the same properties w.r.t. JU- and UA-semantics.

Throughout this section we assume that all programs are finite. We also assume that all non-disjunctive rules π, originating in some DLP, have exactly one literal L in their heads. This latter assumption does not remove any significant expressivity from DLPs under the JU- and UA-semantics since each constraint $(\leftarrow B_\pi.)$ can be equivalently encoded as the rule $(p_\perp \leftarrow \sim p_\perp, B_\pi.)$ where $p_\perp$ is a fresh atom designated for this purpose.

5.1. Programs with Nested Expressions

We adopt the syntax and stable model semantics of logic programs with nested expressions [71]. A *nested rule* is an expression π of the form $(H_\pi \leftarrow B_\pi.)$ where H_π and B_π, called the *head* and *body* of π, are expressions built inductively from the propositional atoms in $\mathcal{A}$ and the 0-place connectives $\perp$ and $\top$ using the unary connective $\sim$ (default negation) and the binary connectives $\wedge$ and $\vee$.[10] A nested rule of the form $(H_\pi \leftarrow \top.)$ is usually identified with the formula H_π. A *nested program* is a finite set of nested rules.

---

[10]Within this section, we refer to such expressions simply as *formulas* even though they are different from the propositional formulas introduced in Section 2.

Each disjunctive program $\mathcal{P}$ and DLP $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$, as defined in Section 2, is translated, respectively, to a nested program and a sequence thereof as follows:

$$\mathcal{P}^{\circledcirc} = \left\{ \left. \bigvee \mathsf{H}_\pi \leftarrow \bigwedge \mathsf{B}_\pi. \ \right| \ \pi \in \mathcal{P} \right\} \qquad \text{and} \qquad \boldsymbol{P}^{\circledcirc} = \langle \mathcal{P}_i^{\circledcirc} \rangle_{i<n} \ .$$

Satisfaction of a formula ϕ in an interpretation J, denoted by $J \models \phi$, is defined in the usual way. Furthermore, J satisfies a nested rule π, denoted by $J \models \pi$, if $J \models \mathsf{B}_\pi$ implies $J \models \mathsf{H}_\pi$, and J satisfies a nested program $\mathcal{P}$, denoted by $J \models \mathcal{P}$, if $J \models \pi$ for all $\pi \in \mathcal{P}$. The *reduct* of a formula ϕ relative to J, denoted by ϕ^J, is obtained by replacing, in ϕ, every maximal occurrence of a formula of the form $\sim \psi$ with $\perp$ if $J \models \psi$ and with $\top$ otherwise. The reducts of a nested rule π and of a nested program $\mathcal{P}$ are, respectively, $\pi^J = (\mathsf{H}_\pi^J \leftarrow \mathsf{B}_\pi^J.)$ and $\mathcal{P}^J = \{ \pi^J \mid \pi \in \mathcal{P} \}$. Finally, an interpretation J is a *stable model* of a nested program $\mathcal{P}$ if it is subset-minimal among the interpretations that satisfy $\mathcal{P}^J$.

5.2. Condensing into a Nested Program

Now we can proceed with the definition of condensing operators $\oplus_{\mathsf{JU}}$ and $\oplus_{\mathsf{UA}}$ for the JU- and UA-semantics. The property that these operators must fulfill is that for any DLP $\boldsymbol{P}$, the stable models of the nested program resulting from applying the operators to $\boldsymbol{P}$ are exactly the JU- and UA-models of $\boldsymbol{P}$, respectively.

Both $\oplus_{\mathsf{JU}}$ and $\oplus_{\mathsf{UA}}$ are defined by utilising the concept of an *activation formula* which captures the condition under which a literal L is derived by some rule in a nested program $\mathcal{U}$. Formally, the *activation formula for L in $\mathcal{U}$* is defined as follows:

$$\alpha_{\mathcal{U}}(L) = \bigvee \left\{ \left. \bigwedge \mathsf{B}_\pi \ \right| \ \pi \in \mathcal{U} \wedge \mathsf{H}_\pi = L \right\} \ .^{[11]}$$

The operator $\oplus_{\mathsf{JU}}$ is based on the following simple idea: When updating a program $\mathcal{P}$ by a program $\mathcal{U}$, each rule π from $\mathcal{P}$ with literal L in its head must be disabled when some rule from $\mathcal{U}$ for the literal complementary to L is activated. This can be achieved by augmenting the body of π with the additional condition $\sim \alpha_{\mathcal{U}}(\overline{L})$. Formally:

Definition 25 (Condensing Operator $\oplus_{\mathsf{JU}}$). A *JU-rule* is a nested rule with a single literal in its head and a *JU-program* is a nested program that consists of JU-rules.

The binary operator $\oplus_{\mathsf{JU}}$ on the set of all JU-programs is defined as follows: Given two JU-programs $\mathcal{P}$ and $\mathcal{U}$, the JU-program $\mathcal{P} \oplus_{\mathsf{JU}} \mathcal{U}$ consists of the following rules:

1. for all $\pi \in \mathcal{P}$, the nested rule $\left(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \sim \alpha_{\mathcal{U}}(\overline{\mathsf{H}_\pi}) . \right)$;

2. all nested rules from $\mathcal{U}$.

In case of the UA-semantics, additional choice rules are needed. Particularly, for every nested rule $\pi \in \mathsf{all}(\boldsymbol{P})^{\circledcirc}$ whose head is an atom, the update operator needs to include the nested rule $(\mathsf{H}_\pi \vee \overline{\mathsf{H}_\pi} \leftarrow \mathsf{B}_\pi.)$ in the resulting program. Intuitively, these additional rules account for the differences in the definition of $\mathsf{rej}_{\mathsf{JU}}(\boldsymbol{P}, J)$ and $\mathsf{rej}_{\mathsf{UA}}(\boldsymbol{P}, J)$ by making sure that no rule is ever completely eliminated, but stays partially in effect by generating alternative worlds for the atom in its head: one where it is satisfied and one where it is not. Essentially, this means that whenever the original body of the rule is satisfied, its head atom cannot be assumed false by default and is interpreted "classically" instead.

---

[11]Note that $\bigvee \emptyset$ is simply $\perp$.

Definition 26 (Condensing Operator $\oplus_{\mathsf{UA}}$)**.** A *UA-rule* is a nested rule with either a single literal or a disjunction of two literals L and $\overline{L}$ in its head, and a *UA-program* is a nested program that consists of UA-rules.

   The binary operator $\oplus_{\mathsf{UA}}$ on the set of all UA-programs is defined as follows: Given two UA-programs $\mathcal{P}$ and $\mathcal{U}$, the UA-program $\mathcal{P} \oplus_{\mathsf{UA}} \mathcal{U}$ consists of the following rules:

1. for all $\pi \in \mathcal{P}$ with $\mathsf{H}_\pi \in \mathcal{L}$, the nested rule $\big(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \sim \alpha_\mathcal{U}(\overline{\mathsf{H}_\pi}) \, . \big)$;

2. all $\pi \in \mathcal{P}$ such that H_π is of the form $L \vee \overline{L}$;

3. for all $\pi \in \mathcal{U}$ with $\mathsf{H}_\pi \in \mathcal{A}$, the nested rule $\big(\mathsf{H}_\pi \vee \overline{\mathsf{H}_\pi} \leftarrow \mathsf{B}_\pi . \big)$;

4. all nested rules from $\mathcal{U}$.

   The following example illustrates the relationship between the JU- and UA-semantics and the condensing operators $\oplus_{\mathsf{JU}}$ and $\oplus_{\mathsf{UA}}$, while the subsequent theorem establishes it in general.

Example 27. Suppose that programs $\mathcal{P}, \mathcal{U}$ and $\mathcal{V}$, written as nested programs, are as follows:

$$
\begin{array}{lll}
\mathcal{P}^\circledcirc: & p \leftarrow \top. & \mathcal{U}^\circledcirc: \quad \sim p \leftarrow \sim q \wedge \sim r. & \mathcal{V}^\circledcirc: \quad p \leftarrow s. \\
& q \leftarrow p. & \phantom{\mathcal{U}^\circledcirc:}\quad \sim p \leftarrow s. & \phantom{\mathcal{V}^\circledcirc:}\quad r \leftarrow r. \\
& r \leftarrow \top. & \phantom{\mathcal{U}^\circledcirc:}\quad \sim r \leftarrow \top. & \phantom{\mathcal{V}^\circledcirc:}\quad s \leftarrow \top.
\end{array}
$$

In addition to the rules from $\mathcal{U}^\circledcirc$, the program $\bigoplus_{\mathsf{JU}}\langle \mathcal{P}^\circledcirc, \mathcal{U}^\circledcirc \rangle$ contains the following three nested rules:

$$
p \leftarrow \top \wedge \sim((\sim q \wedge \sim r) \vee s). \qquad q \leftarrow p \wedge \sim\bot. \qquad r \leftarrow \top \wedge \sim\top. \tag{6}
$$

Also, the program $\bigoplus_{\mathsf{UA}}\langle \mathcal{P}^\circledcirc, \mathcal{U}^\circledcirc \rangle$ consists of the rules in $\bigoplus_{\mathsf{JU}}\langle \mathcal{P}^\circledcirc, \mathcal{U}^\circledcirc \rangle$ and of three additional choice rules. Note that these rules are not part of the program $\mathcal{P}^\circledcirc \oplus_{\mathsf{UA}} \mathcal{U}^\circledcirc$. They belong to the program $\emptyset \oplus_{\mathsf{UA}} \mathcal{P}^\circledcirc$, and, consequently, also to $(\emptyset \oplus_{\mathsf{UA}} \mathcal{P}^\circledcirc) \oplus_{\mathsf{UA}} \mathcal{U}^\circledcirc = \bigoplus_{\mathsf{UA}}\langle \mathcal{P}^\circledcirc, \mathcal{U}^\circledcirc \rangle$. They are as follows:

$$
p \vee \sim p \leftarrow \top. \qquad\qquad q \vee \sim q \leftarrow p. \qquad r \vee \sim r \leftarrow \top. \tag{7}
$$

Furthermore, both $\bigoplus_{\mathsf{JU}}\langle \mathcal{P}^\circledcirc, \mathcal{U}^\circledcirc \rangle$ and $\bigoplus_{\mathsf{UA}}\langle \mathcal{P}^\circledcirc, \mathcal{U}^\circledcirc \rangle$ have the stable models $\emptyset$ and $\{\, p, q \,\}$ which coincide with the JU- and UA-models of the DLP $\langle \mathcal{P}, \mathcal{U} \rangle$.

   The situation is more interesting after $\mathcal{V}^\circledcirc$ is added to the update sequence. In addition to the rules from $\mathcal{V}^\circledcirc$, the program $\bigoplus_{\mathsf{JU}}\langle \mathcal{P}^\circledcirc, \mathcal{U}^\circledcirc, \mathcal{V}^\circledcirc \rangle$ contains the following rules:

$$
\begin{array}{lll}
p \leftarrow \top \wedge \sim((\sim q \wedge \sim r) \vee s) \wedge \sim\bot. & q \leftarrow p \wedge \sim\bot \wedge \sim\bot. & r \leftarrow \top \wedge \sim\top \wedge \sim\bot. \\
\sim p \leftarrow \sim q \wedge \sim r \wedge \sim s. & \sim p \leftarrow s \wedge \sim s. & \sim r \leftarrow \top \wedge \sim r.
\end{array}
$$

Also, the program $\bigoplus_{\mathsf{UA}}\langle \mathcal{P}^\circledcirc, \mathcal{U}^\circledcirc, \mathcal{V}^\circledcirc \rangle$ consists of the rules in $\bigoplus_{\mathsf{JU}}\langle \mathcal{P}^\circledcirc, \mathcal{U}^\circledcirc, \mathcal{V}^\circledcirc \rangle$, the three choice rules listed in (7) and, additionally, the following three choice rules rules originating in $\mathcal{V}^\circledcirc$:

$$
p \vee \sim p \leftarrow s. \qquad\qquad r \vee \sim r \leftarrow r. \qquad\qquad s \vee \sim s \leftarrow \top.
$$

   Note that some body conjuncts, such as $\sim\bot$, and even whole rules, for instance $(\sim p \leftarrow s \wedge \sim s.)$, can can be eliminated from the resulting programs without affecting their stable models or the stable models resulting from further updates. Slightly less elegant definitions of $\oplus_{\mathsf{JU}}$ and $\oplus_{\mathsf{UA}}$ could be used to perform such simplifications automatically. For illustration, Figure 2 lists the simplified versions of the nested programs $\bigoplus_{\mathsf{JU}}\langle \mathcal{P}^\circledcirc, \mathcal{U}^\circledcirc, \mathcal{V}^\circledcirc \rangle$ and $\bigoplus_{\mathsf{UA}}\langle \mathcal{P}^\circledcirc, \mathcal{U}^\circledcirc, \mathcal{V}^\circledcirc \rangle$. It is also not difficult to verify that the unique stable model of the former program is $\{\, p, q, s \,\}$, which coincides with the unique JU-model of $\langle \mathcal{P}, \mathcal{U}, \mathcal{V} \rangle$. On the other hand, due to the rule $(r \vee \sim r \leftarrow \top.)$, the latter program has the additional stable model $\{\, p, q, r, s \,\}$, which coincides with the additional UA-model of the DLP $\langle \mathcal{P}, \mathcal{U}, \mathcal{V} \rangle$.

$p \leftarrow \sim((\sim q \wedge \sim r) \vee s).$		$q \leftarrow p.$	$p \leftarrow \sim((\sim q \wedge \sim r) \vee s).$	$q \leftarrow p.$
$\sim p \leftarrow \sim q \wedge \sim r \wedge \sim s.$			$\sim p \leftarrow \sim q \wedge \sim r \wedge \sim s.$	
$p \leftarrow s.$		$s \leftarrow \top.$	$p \leftarrow s.$	$s \leftarrow \top.$
			$p \vee \sim p \leftarrow \top.$	$q \vee \sim q \leftarrow p.$
			$r \vee \sim r \leftarrow \top.$	$s \vee \sim s \leftarrow \top.$

Figure 2: Nested programs $\bigoplus_{\mathrm{JU}} \langle \mathcal{P}^\circledcirc, \mathcal{U}^\circledcirc, \mathcal{V}^\circledcirc \rangle$ (left) and $\bigoplus_{\mathrm{UA}} \langle \mathcal{P}^\circledcirc, \mathcal{U}^\circledcirc, \mathcal{V}^\circledcirc \rangle$ (right) without irrelevant rules and body conjuncts.

Theorem 28 (State Condensing Using $\oplus_{\mathrm{JU}}$ and $\oplus_{\mathrm{UA}}$)**.** *Let $\boldsymbol{P}$ be a DLP. An interpretation J is a*

(i) *JU-model of $\boldsymbol{P}$ if and only if it is a stable model of $\bigoplus_{\mathrm{JU}} \boldsymbol{P}^\circledcirc$;*

(ii) *UA-model of $\boldsymbol{P}$ if and only if it is a stable model of $\bigoplus_{\mathrm{UA}} \boldsymbol{P}^\circledcirc$.*

Proof. See Appendix C, page 55. $\qquad \square$

The significance of this theorem is in that it demonstrates that the operators $\oplus_{\mathrm{JU}}$ and $\oplus_{\mathrm{UA}}$ provide a full characterisation of the JU- and UA-semantics: They condense any DLP $\boldsymbol{P}$ into a single nested program $\mathcal{P}$ such that the stable models of $\mathcal{P}$ coincide with the JU- and UA-models of $\boldsymbol{P}$, respectively, and any further updates of $\boldsymbol{P}$ can be equivalently performed directly on $\mathcal{P}$ using the operators.

Note that since the operators manipulate rules on the syntactic level, they are completely faithful to JU- and UA-semantics, even when the DLP contains local cycles (see also Theorem 23 where local cycles form a special case due to the fact that exception functions manipulate rules on the semantic level).

Interestingly, an operator very similar to $\oplus_{\mathrm{JU}}$ has been studied by Osorio and Cuevas [55]. They have proven that it characterises the UA-semantics for the case of a single update but did not consider the possibility of performing iterated updates. Their result is also a consequence of Theorem 28 and of the fact that the JU- and UA-semantics provide the same result when only a single update is performed.

5.3. Condensing into a Disjunctive Program

The condensing operators defined in the previous section can be further refined in order to produce a program that meets additional requirements. In the present section we show that nested expressions can be completely eliminated from the resulting program while still preserving the same tight relationship with the original rule update semantics. Thus, we introduce an additional pair of operators, $\oplus_{\mathrm{JU}}^\vee$ and $\oplus_{\mathrm{UA}}^\vee$, that operate on disjunctive programs with default negation in heads of rules. Note that due to the non-minimality of JU- and UA-models for certain DLPs, disjunctive programs *without* default negation in heads of rules would already be insufficient for this purpose.

The ideas underlying the new operators are fairly straightforward. Essentially, nested expressions are introduced into the resulting programs only by the negations of activation formulas in their bodies, so these are the parts of rules that need to be translated into conjunctions in bodies and disjunctions in heads of rules. In particular, by utilising De Morgan's law and distributivity of conjunction over disjunction, we can obtain a new formula, a disjunction of conjunctions of default literals and double-negated atoms, that is strongly equivalent to the original formula. For instance, in case of the first rule in (6), we can equivalently write the condition $\sim((\sim q \wedge \sim r) \vee s)$ as $(\sim\sim q \wedge \sim s) \vee (\sim\sim r \wedge \sim s)$. Then it suffices to break up the resulting rule into multiple rules, each with one of the disjuncts of this formula in the body, and remove one of the negations from each double-negated literals and "move" it into the head of the newly constructed rule. In case of the first rule in (6), the result would be the disjunctive rules

$$p \vee \sim q \leftarrow \sim s. \qquad \text{and} \qquad p \vee \sim r \leftarrow \sim s. \qquad (8)$$

We call each set of literals, without double negation, within each of the disjuncts described above a *blocking set*. Formally, if the activation formula $\alpha_{\mathcal{U}\odot}(L)$ contains $\top$ as one of its disjuncts, then there is no blocking set for L in $\mathcal{U}$. Otherwise, suppose that $\alpha_{\mathcal{U}\odot}(L) = (L_1^1 \wedge \cdots \wedge L_{k_1}^1) \vee \cdots \vee (L_1^n \wedge \cdots \wedge L_{k_n}^n)$. A *blocking set for L in $\mathcal{U}$* is any set of literals $\left\{ \overline{L_{i_1}^1}, \ldots, \overline{L_{i_n}^n} \right\}$ where $1 \le i_j \le k_j$ for every j with $1 \le j \le n$. We denote the set of all blocking sets for L in $\mathcal{U}$ by $\beta_{\mathcal{U}}(L)$.

Each nested rule $\left(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \sim \alpha_{\mathcal{U}\odot}(\overline{\mathsf{H}_\pi}) \, . \right)$ can thus be replaced by the set of disjunctive rules

$$\left\{ \left. \mathsf{H}_\pi ; \overline{S^+} \leftarrow \mathsf{B}_\pi, \overline{S^-}. \; \right| \; S \in \beta_{\mathcal{U}}\left(\overline{\mathsf{H}_\pi}\right) \right\} \; .$$

Furthermore, when H_π is a default literal, it is more convenient to move the new default literals from the head into the body since this operation preserves stable models [38] and makes it easier to pinpoint the original head literal in the rule. This leads us to the following definition of $\oplus_{\mathsf{JU}}^\vee$:

Definition 29 (Condensing Operator $\oplus_{\mathsf{JU}}^\vee$). Given a disjunctive rule π, a literal L is the *original head of π* if either $L \in \mathcal{A}$ and $\mathsf{H}_\pi \cap \mathcal{A} = \{L\}$, or $L \notin \mathcal{A}$ and $\mathsf{H}_\pi = \{L\}$.

The binary operator $\oplus_{\mathsf{JU}}^\vee$ on the set of all disjunctive programs is defined as follows: Given two disjunctive programs $\mathcal{P}$ and $\mathcal{U}$, the disjunctive program $\mathcal{P} \oplus_{\mathsf{JU}}^\vee \mathcal{U}$ consists of the following rules:

1. for all $\pi \in \mathcal{P}$ with original head $L \in \mathcal{A}$ and all $S \in \beta_{\mathcal{U}}(\overline{L})$, the rule $\left(\mathsf{H}_\pi ; \overline{S^+} \leftarrow \mathsf{B}_\pi, \overline{S^-}. \right)$;

2. for all $\pi \in \mathcal{P}$ with original head $L \notin \mathcal{A}$ and all $S \in \beta_{\mathcal{U}}(\overline{L})$, the rule $\left(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi, S^+, \overline{S^-}. \right)$;

3. all rules from $\mathcal{U}$.

As for the UA-semantics, similar modifications can be applied in the definition of $\oplus_{\mathsf{UA}}$ to obtain an operator that produces a disjunctive program. Furthermore, due to the additional choice rules included in the result, the rules can be further simplified, when compared to the rules produced by $\oplus_{\mathsf{JU}}^\vee$. In particular, the first group of rules can be treated the same way as the second, leading to the following definition of $\oplus_{\mathsf{UA}}^\vee$:

Definition 30 (Condensing Operator $\oplus_{\mathsf{UA}}^\vee$). The binary operator $\oplus_{\mathsf{UA}}^\vee$ on the set of all disjunctive programs is defined as follows: Given two disjunctive programs $\mathcal{P}$ and $\mathcal{U}$, the disjunctive program $\mathcal{P} \oplus_{\mathsf{UA}}^\vee \mathcal{U}$ consists of the following rules:

1. for all $\pi \in \mathcal{P}$ with $\mathsf{H}_\pi = \{L\}$ and all $S \in \beta_{\mathcal{U}}(\overline{L})$, the rule $\left(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi, S^+, \overline{S^-}. \right)$;

2. all $\pi \in \mathcal{P}$ such that H_π is of the form $\{p, \sim p\}$ for some $p \in \mathcal{A}$;

3. for all $\pi \in \mathcal{U}$ with $\mathsf{H}_\pi = \{p\}$ for some $p \in \mathcal{A}$, the rule $(p; \sim p \leftarrow \mathsf{B}_\pi.)$;

4. all rules from $\mathcal{U}$.

If we consider the programs $\mathcal{P}$, $\mathcal{U}$ and $\mathcal{V}$ from Example 27, the two main differences between the disjunctive program $\bigoplus_{\mathsf{JU}}^\vee \langle \mathcal{P}, \mathcal{U}, \mathcal{V} \rangle$ and its nested counterpart $\bigoplus_{\mathsf{JU}} \langle \mathcal{P}, \mathcal{U}, \mathcal{V} \rangle$, besides their different syntax, are as follows: 1) the nested rule is turned into the two disjunctive rules listed in (8), and 2) the rule $r \leftarrow \top \wedge \sim \top \wedge \sim \bot$ has no counterpart in the disjunctive program as there is no blocking set for $\sim r$ in $\mathcal{U}$. The differences between $\bigoplus_{\mathsf{UA}}^\vee \langle \mathcal{P}, \mathcal{U}, \mathcal{V} \rangle$ and $\bigoplus_{\mathsf{UA}} \langle \mathcal{P}, \mathcal{U}, \mathcal{V} \rangle$ are analogical and in Figure 3 we list the disjunctive programs simplified in the same way as their nested counterparts in Figure 2.

From a general perspective, the operators $\oplus_{\mathsf{JU}}^\vee$ and $\oplus_{\mathsf{UA}}^\vee$ preserve the main property of $\oplus_{\mathsf{JU}}$ and $\oplus_{\mathsf{UA}}$.

Theorem 31 (State Condensing Using $\oplus_{\mathsf{JU}}^\vee$ and $\oplus_{\mathsf{UA}}^\vee$). *Let $\boldsymbol{P}$ be a DLP. An interpretation J is a*

(i) JU-model of $\boldsymbol{P}$ if and only if it is a stable model of $\bigoplus_{\mathsf{JU}}^\vee \boldsymbol{P}$;

$p; \sim q \leftarrow \sim s.$	$q \leftarrow p.$	$p \leftarrow q, \sim s.$	$q \leftarrow p.$
$p; \sim r \leftarrow \sim s.$		$p \leftarrow r, \sim s.$	
$\sim p \leftarrow \sim q, \sim r, \sim s.$		$\sim p \leftarrow \sim q, \sim r, \sim s.$	
$p \leftarrow s.$	$s.$	$p \leftarrow s.$	$s.$
		$p; \sim p.$	$q; \sim q \leftarrow p.$
		$r; \sim r.$	$s; \sim s.$

Figure 3: Disjunctive programs $\bigoplus_{\mathsf{JU}}^{\vee}\langle \mathcal{P}, \mathcal{U}, \mathcal{V}\rangle$ (left) and $\bigoplus_{\mathsf{UA}}^{\vee}\langle \mathcal{P}, \mathcal{U}, \mathcal{V}\rangle$ (right) without irrelevant rules.

(ii) UA-model of $\boldsymbol{P}$ if and only if it is a stable model of $\bigoplus_{\mathsf{UA}}^{\vee} \boldsymbol{P}$.

Proof. See Appendix C, page 58. □

Although operators $\oplus_{\mathsf{JU}}^{\vee}$ and $\oplus_{\mathsf{UA}}^{\vee}$ eliminate the necessity for using nested rules to condense a DLP into a single program, this comes at a cost. Namely, the size of the nested program resulting from applying operators $\oplus_{\mathsf{JU}}$ and $\oplus_{\mathsf{UA}}$ is always linear in size of the argument programs, while in case of $\oplus_{\mathsf{JU}}^{\vee}$ and $\oplus_{\mathsf{UA}}^{\vee}$, the resulting program can be exponentially larger. Furthermore, Figures 2 and 3 suggest that the representations produced by $\oplus_{\mathsf{JU}}^{\vee}$ and $\oplus_{\mathsf{UA}}^{\vee}$ will be less faithful to the form of the rules in the original programs, and thus less readable. This indicates that the nested program is more suitable as a way to store the condensed program, both in terms of space and readability. Additionally, in order to find its stable models, a more efficient translation can be used that utilises additional meta-level atoms to prevent the exponential explosion (see [24, 48] for further details). However, such a translation will no longer be equivalent to the original program sequence w.r.t. performing further updates.

6. Abstract Exception-Based Operators

In this section we generalise exception-based operators to arbitrary knowledge representation formalisms with monotonic model-theoretic semantics.

Thus, we assume to be using some knowledge representation formalism in which a *knowledge base* is a subset of the set of all *knowledge atoms* Ω and $\mathcal{Z}$ denotes the set of all *semantic structures* among which the *models* of knowledge atoms are chosen. The set of models of a knowledge atom α is denoted by $[\![\alpha]\!]$. The *semantic characterisation* of a knowledge base $\mathcal{K}$ is the *set of sets of models* of its knowledge atoms: $\langle\!\langle \mathcal{K} \rangle\!\rangle = \{\, [\![\alpha]\!] \mid \alpha \in \mathcal{K} \,\}$. The models of $\mathcal{K}$ are the models of all its elements, i.e. $[\![\mathcal{K}]\!] = \bigcap \langle\!\langle \mathcal{K} \rangle\!\rangle$.

An abstract exception-based update operator is characterised by an *exception function* that, given the set of models of a knowledge atom α and the semantic characterisations of the original and updating knowledge base, returns the semantic structures that are to be introduced as exceptions to α.

Definition 32 (Exception Function). An *exception function* is any function $\varepsilon : 2^{\mathcal{Z}} \times 2^{2^{\mathcal{Z}}} \times 2^{2^{\mathcal{Z}}} \to 2^{\mathcal{Z}}$.

Given such an exception function and knowledge bases $\mathcal{K}$ and $\mathcal{U}$, it naturally follows that the semantic characterisation resulting from updating $\mathcal{K}$ by $\mathcal{U}$ should consist of sets of models of each knowledge atom α from $\mathcal{K}$, each augmented with the respective exceptions, and also the unmodified sets of models of knowledge atoms from $\mathcal{U}$. In other words, we obtain the set of sets of models

$$\{\, [\![\alpha]\!] \cup \varepsilon([\![\alpha]\!], \langle\!\langle \mathcal{K} \rangle\!\rangle, \langle\!\langle \mathcal{U} \rangle\!\rangle) \mid \alpha \in \mathcal{K} \,\} \cup \langle\!\langle \mathcal{U} \rangle\!\rangle \ . \tag{9}$$

27

Turning to the syntactic side, an *update operator* is binary function over 2^Ω that takes the original knowledge base and its update as inputs and returns the updated knowledge base. An *exception-based update operator* is then formalised as follows:

Definition 33 (Abstract Exception-Based Update Operator). Given an exception function ε, an update operator $\oplus$ is ε-*based* if for all $\mathcal{K}, \mathcal{U} \subseteq \Omega$, $\langle\!\langle \mathcal{K} \oplus \mathcal{U} \rangle\!\rangle_{\text{RE}}$ is equal to (9). Also, $\oplus$ is *exception-based* if it is ε-based for some exception function ε.

7. Belief Updates Using Exception-Based Operators

Concrete exception-based operators for propositional knowledge bases are obtained from the abstract framework presented in the previous section by identifying the set of knowledge atoms Ω with the set of all propositional formulas and the set of semantic structures $\mathcal{Z}$ with propositional interpretations.

This setup allows us to immediately prove that most conceivable model-based operators can be faithfully modelled as exception-based ones. Particularly, this is possible for any update operator satisfying the uncontroversial postulates (B1), (B2.1) and (B4) [34].

Theorem 34 (Model-Based Updates Using Exception-Based Operators). *If $\diamond$ is a belief update operator that satisfies (B1), (B2.1) and (B4), then there exists an exception function ε such that for every ε-based update operator $\oplus$ and all finite sequences of knowledge bases $\mathcal{D}$, $[\![\diamond \mathcal{D}]\!] = [\![\bigoplus \mathcal{D}]\!]$.*

Proof. See Appendix E, page 59. □

An interesting point regarding this result is that the principles (B1), (B2.1) and (B4) are not specific to update operators, they are also satisfied by *AGM revision* operators. These operators are developed for the case of revising a *belief set* which is a set of formulas closed w.r.t. a logical consequence operator Cn. A revision operator $\star$ takes an original belief set $\mathcal{T}$ and a formula μ representing its revision and produces the revised belief set $\mathcal{T} \star \mu$. The typical properties satisfied by AGM revision operators include *success*, *inclusion* and *extensionality* [33], formalised, respectively, as

$$\mu \in \mathcal{T} \star \mu \ , \qquad \mathcal{T} \star \mu \subseteq Cn(\mathcal{T} \cup \{\mu\}) \ , \qquad \text{If } \mu \equiv \nu, \text{ then } \mathcal{T} \star \mu = \mathcal{T} \star \nu.$$

These three properties directly imply that (B1), (B2.1) and (B4) are satisfied by AGM revision operators. Hence, Theorem 34 directly applies to AGM revision operators as well. Note also that the operator adopted for ABox updates in [49], inspired by WIDTIO, performs a deductive closure of the ABox before updating it, so it corresponds to the standard *full meet AGM revision operator*.

Turning to formula-based belief update operators, we can achieve similar results by introducing the following principles satisfied by many formula-based operators. Here, for any knowledge base $\mathcal{B}$, $\langle\!\langle \mathcal{B} \rangle\!\rangle^{\mathcal{I}}$ denotes the set $\langle\!\langle \mathcal{B} \rangle\!\rangle \cup \{\mathcal{I}\}$. The principles are as follows:

(F1) $\langle\!\langle \mathcal{B} \circ \mathcal{U} \rangle\!\rangle \supseteq \langle\!\langle \mathcal{U} \rangle\!\rangle$.

(F2.1) $\langle\!\langle \mathcal{B} \cup \mathcal{U} \rangle\!\rangle \supseteq \langle\!\langle \mathcal{B} \circ \mathcal{U} \rangle\!\rangle$.

(F4)   If $\langle\!\langle \mathcal{B} \rangle\!\rangle^{\mathcal{I}} = \langle\!\langle \mathcal{C} \rangle\!\rangle^{\mathcal{I}}$ and $\langle\!\langle \mathcal{U} \rangle\!\rangle^{\mathcal{I}} = \langle\!\langle \mathcal{V} \rangle\!\rangle^{\mathcal{I}}$, then $\langle\!\langle \mathcal{B} \circ \mathcal{U} \rangle\!\rangle^{\mathcal{I}} = \langle\!\langle \mathcal{C} \circ \mathcal{V} \rangle\!\rangle^{\mathcal{I}}$.

We can see that (F1) and (F2.1) are *stronger* versions of (B1), and (B2.1), respectively. While (F1) requires that the sets of models of formulas in $\mathcal{U}$ be retained in the semantic characterisation of $\mathcal{B} \circ \mathcal{U}$, (F2.1) states that every formula in $\mathcal{B} \circ \mathcal{U}$ be equivalent to some formula in $\mathcal{B} \cup \mathcal{U}$. Intuitively, this means that $\mathcal{B} \circ \mathcal{U}$ is obtained from $\mathcal{B} \cup \mathcal{U}$ by deleting some of its elements, modulo equivalence. Finally, (F4) is a reformulation of (B4) that is satisfied by formula-based operators – it can be seen as syntax-independence w.r.t. the set of

sets of models of a knowledge base, modulo the presence of tautologies, instead of the overall set of models as in (B4). In some ways it is *weaker* than (B4) as its antecedent is much stronger.

The WIDTIO operator [30, 31, 73] satisfies all of these principles, and so does the Bold operator [13] if it is based on a remainder selection function that selects remainders with the same semantic characterisation when given sets of remainders with the same sets of semantic characterisations. More formally:

Definition 35 (Regular Bold Operator). For any set of remainders $\mathcal{R}$ we define $((\mathcal{R}))^{\mathcal{I}} = \{ \langle\!\langle \mathcal{B} \rangle\!\rangle^{\mathcal{I}} \mid \mathcal{B} \in \mathcal{R} \}$. We say that the Bold operator $\circ_{\mathsf{bold}}^{s}$ is *regular* if for all sets of remainders $\mathcal{R}_1$, $\mathcal{R}_2$ such that $((\mathcal{R}_1))^{\mathcal{I}} = ((\mathcal{R}_2))^{\mathcal{I}}$ it holds that $\langle\!\langle s(\mathcal{R}_1) \rangle\!\rangle^{\mathcal{I}} = \langle\!\langle s(\mathcal{R}_2) \rangle\!\rangle^{\mathcal{I}}$.

The regularity condition guarantees a certain degree of independence of syntax, e.g. given the sets of remainders $\mathcal{R}_1 = \{\, \{\, p \,\}, \{\, q \,\} \,\}$ and $\mathcal{R}_2 = \{\, \{\, p \wedge p \,\}, \{\, q \vee q \,\} \,\}$, a regular Bold operator either selects $\{\, p \,\}$ from $\mathcal{R}_1$ and $\{\, p \wedge p \,\}$ from $\mathcal{R}_2$, or it selects $\{\, q \,\}$ from $\mathcal{R}_1$ and $\{\, q \vee q \,\}$ from $\mathcal{R}_2$. A non-regular one might select, say, $\{\, p \,\}$ from $\mathcal{R}_1$ and $\{\, q \vee q \,\}$ from $\mathcal{R}_2$. Thus the regularity condition ensures that the operator is independent of the syntax of individual formulas in the knowledge base.

The Cross-Product operator [30] satisfies (F1) (thus also (B1)), (B2.1) and (F4), but not (F2.1).

Proposition 36 (Properties of Formula-Based Updates). *The WIDTIO and regular Bold operators satisfy (F1), (F2.1) and (F4). The Cross-Product operator satisfies (F1), (B2.1) and (F4) but does not satisfy (F2.1).*

Proof. See Appendix E, page 62. □

The following result establishes that formula-based operators such as WIDTIO and regular Bold can be fully captured by exception-based operators. In addition, operators such as Cross-Product can be captured for the case of a single update.

Theorem 37 (Formula-Based Updates Using Exception-Based Operators). *If $\circ$ is an update operator that satisfies (F1), (F2.1) and (F4), then there exists an exception function ε such that for every ε-based update operator $\oplus$ and all finite sequences of knowledge bases $\mathcal{D}$, $[\![\bigcirc \mathcal{D}]\!] = [\![\bigoplus \mathcal{D}]\!]$.*

If $\circ$ is an update operator that satisfies (B1), (B2.1) and (F4), then there exists an exception function ε such that for every ε-based update operator $\oplus$ and all knowledge bases $\mathcal{B}, \mathcal{U}$, $[\![\mathcal{B} \circ \mathcal{U}]\!] = [\![\mathcal{B} \oplus \mathcal{U}]\!]$.

Proof. See Appendix E, page 63. □

Similarly as with principles of model-based update operators, principles (F1), (F2.1) and (F4) are closely related with the properties of *base revision operators* [27, 33]. In particular, two types of base revision are identified in [33], the *internal* and *external base revision*. Both of them satisfy base revision counterparts of *success* and *inclusion* and, in addition, internal revision operators satisfy a property called *uniformity*. These three principles together entail that internal revision operators satisfy (F1), (F2.1) and one half of (F4); the other half can be achieved by putting additional constraints on the two-place selection function that generates the revision operator, similar to the *regularity* condition we imposed on the Bold operator above. Such regular internal revision operators are thus directly subject to Theorem 37. The same however does not hold for regular external revision operators as they need not satisfy *uniformity*. Note also that the WIDTIO and Bold operators coincide with *internal full meet base revision* and *internal maxichoice base revision* operators, respectively.

8. Conclusion

Throughout this paper we developed a novel perspective on knowledge updates and demonstrated that it offers the first unifying ground for characterising rule update semantics as well as both model- and formula-based classical update operators.

More particularly, we defined a new monotonic characterisation of rules, the RE-models, and introduced a generic method for specifying semantic rule update operators in which a logic program is viewed as the *set of sets of RE-models of its rules* and updates are performed by introducing additional interpretations to the sets of RE-models of rules in the original program. This framework allowed us to define concrete semantic rule update operators that enjoy an interesting combination of syntactic as well as semantic properties that had never been reconciled before. Furthermore, it turned out that these operators can faithfully characterise the foundational JU- and UA-semantics for rule updates.

These insights also allowed us to address the long enduring problem of *state condensing*, i.e. transforming a sequence of answer-set programs – interpreted as updates – into a single answer-set program written in the same alphabet. Partly, this problem emerges because some rule update semantics employ complex mechanisms such as preferences or other minimality criteria that make it impossible to encode the result in a single answer-set program. Others have model-theoretic characterisations that assign non-minimal models to certain update sequences, and it is well known that stable models of non-disjunctive answer-set programs are minimal. By resorting to more expressive classes of answer-set programs, namely nested and disjunctive, we solved this problem for both JU- and UA-semantics. In all four cases, two for each semantics using both classes of answer-set programs, the resulting program is written with the same alphabet and is ready to be further updated. We have illustrated with some examples that the resulting programs written using nested answer-set programming are perhaps more readable than those written using disjunctive answer-set programs, in the sense that they more closely match the intuitions underlying the semantics for updates that we consider.

Furthermore, we defined *abstract exception-based operators* that can be used in any knowledge representation formalism with a monotonic model-theoretic semantics. Then we showed that exception-based operators for propositional knowledge bases can fully capture update operators that form the basis of ontology updates, such as the model-based Winslett's operator, or the formula-based WIDTIO and Bold operators [13, 19, 49, 51]. The Cross-Product operator can be captured when a single update is performed and the same can be said about the Set-Of-Theories operator since for a single update it is equivalent to the Cross-Product operator [73]. Nevertheless, neither of these two operators offers a viable alternative for updating ontologies – Cross-Product requires that disjunctions of ontology axioms be performed, which is typically not supported in DLs, and Set-Of-Theories produces a disjunctive ontology which is impractical and deviates from mainstream DL research.

Overall, exception functions and exception-based operators can capture both traditional syntax-based approaches to rule updates as well as a wide range of model- and formula-based belief update operators. Thus, they offer a uniform framework that bridges two very distinct approaches to updates, previously considered irreconcilable. These findings are essential to better understand their interrelations. In addition, they open up new possibilities for addressing updates of Hybrid Knowledge Bases consisting of both an ontology and a rule component since the different methods used for dealing with ABox, TBox and rule updates can be viewed uniformly by looking at their associated exception functions. When coupled with a counterpart of SE- or RE-models in the context of hybrid knowledge bases, this can lead to universal hybrid update semantics which, in turn, can further improve our understanding of the distinct update paradigms.

Our investigation also directly points to challenges that need to be tackled next. First, semantic characterisations of additional rule update semantics need to be investigated. This poses a number of challenges

due to the need to detect non-tautological irrelevant updates [5, 61, 62]. For instance, the simple functions examined in this paper, as well as the original JU- and UA-semantics, cannot distinguish an update of $\{\,p.\,\}$ by $\mathcal{U} = \{\,\sim p \leftarrow \sim q., \sim q \leftarrow \sim p.\,\}$, where it is plausible to introduce the exception $(\emptyset, \emptyset)$ (and the stable model $\emptyset$ along with it), from an update of $\{\,p., q.\,\}$ by $\mathcal{U}$, where such an exception should not be introduced due to the cyclic dependency of justifications to reject $(p.)$ and $(q.)$. In such situations, context-aware functions need to be used. Such functions have the potential of satisfying properties such as (P3) and (Associativity). They would facilitate the search for condensing operators for other rule update semantics and perhaps shed some light on the problem of *updating disjunctive programs* which has received very little attention up until now.

Another challenge is to find additional logical characterisations of rules, namely a notion of program equivalence that is weaker than RR-equivalence but stronger than RE-equivalence so that both (P4) and properties such as (P2.1) can be achieved under a single notion of program equivalence. In this context, the close relationship between RE-models and *T-models* [74], used in the context of *forgetting* in logic programs, asks for more attention as well.

Computational properties of different classes of exception-based update operators should also be investigated and it might be interesting to look for constrained classes of exception functions that satisfy syntax-independence w.r.t. SR-equivalence. Such functions, however, will not be able to respect causal rejection because SE-models cannot distinguish abolishing rules.

Our discussion of the expressivity of exception-based operators w.r.t. *revision* operators, on both belief sets and belief bases, can be used to tackle and unify approaches to *ontology revision* [32, 57, 58]. This seems relevant even in the context of ontology updates since it has been argued in the literature that the strict distinction between revision and update is not suitable in the context of ontologies [13].

Acknowledgements

Appendix A. Proofs: Robust Equivalence Models

In this section we prove formal properties of RE-models (c.f. Definition 7). We begin by defining a set of representatives of rule equivalence classes induced by SE-models, as it was introduced in [65] since it is needed in our proofs.

Definition 38 (Transformation into an SE-Canonical Rule [65]). Given a rule π, we define the SE-canonical rule $\mathsf{can}_{\mathsf{SE}}(\pi)$ as follows:

    (i) If any of the sets $\mathsf{H}_\pi^+ \cap \mathsf{B}_\pi^+$, $\mathsf{H}_\pi^- \cap \mathsf{B}_\pi^-$ and $\mathsf{B}_\pi^+ \cap \mathsf{B}_\pi^-$ is non-empty, then $\mathsf{can}_{\mathsf{SE}}(\pi) = \tau$.

    (ii) If (i) does not apply and $\mathsf{H}_\pi^+ \setminus \mathsf{B}_\pi^- \neq \emptyset$, then $\mathsf{can}_{\mathsf{SE}}(\pi) = \left((\mathsf{H}_\pi^+ \setminus \mathsf{B}_\pi^-); \overline{(\mathsf{H}_\pi^- \setminus \mathsf{B}_\pi^+)} \leftarrow \mathsf{B}_\pi^+, \overline{\mathsf{B}_\pi^-}. \right)$.

    (iii) If (i) does not apply and $\mathsf{H}_\pi^+ \setminus \mathsf{B}_\pi^- = \emptyset$, then $\mathsf{can}_{\mathsf{SE}}(\pi)$ is the constraint $\left(\leftarrow (\mathsf{B}_\pi^+ \cup \mathsf{H}_\pi^-), \overline{\mathsf{B}_\pi^-}. \right)$.

Proposition 39 ([65]). *Every rule π is SE-equivalent to the SE-canonical rule $\mathsf{can}_{\mathsf{SE}}(\pi)$.*

Corollary 40 ([65]). *No two different SE-canonical rules are SE-equivalent.*

The following presentation follows a similar pattern as the one used in [65] for SE-models. In particular, we introduce a set of representatives of rule equivalence classes induced by RE-models and show how to reconstruct a representative from the set of its RE-models. Then we prove the properties of RE-models that have been formulated in Section 3.

Remark 41. We use the following additional notation: For any rule π, π^+ denotes the rule $(\mathsf{H}_\pi^+ \leftarrow \mathsf{B}_\pi^+.)$ and π^- denotes the rule $\left(\overline{\mathsf{H}_\pi^-} \leftarrow \overline{\mathsf{B}_\pi^-}.\right)$. Note that the definition of an RE-model (c.f. Definition 7) implies that $(I, J) \in [\![\pi]\!]_{\mathsf{RE}}$ if and only if $I \models \pi^+$ or $J \models \pi^-$. This fact is used implicitly in the following proofs.

Lemma 42. *Rules of the following forms are RE-tautological:*

$$p; H \leftarrow p, B. \qquad\qquad H; \sim p \leftarrow B, \sim p. \qquad\qquad H \leftarrow B, p, \sim p.$$

Proof. First assume that a rule π is of the first form and take some $(I, J) \in \mathfrak{X}$. We need to show that (I, J) is an RE-model of π, so it suffices to show that $I \models \pi^+$. This follows from the fact that p belongs to both H_π^+ and B_π^+.

Now suppose that π is of the second form. Given some $(I, J) \in \mathfrak{X}$, we see that the atom p belongs to both H_π^- and B_π^-, so $J \models \pi^-$. Hence, (I, J) is an RE-model of π.

Finally, suppose that π takes the third form and take some $(I, J) \in \mathfrak{X}$. If $J \models \pi^-$, then (I, J) is an RE-model of π. On the other hand, if $J \not\models \pi^-$, then $J \models \sim p$ and, consequently, $I \not\models p$ because I is a subset of J. This implies that $I \models \pi^+$, so we can once again conclude that (I, J) is an RE-model of π. $\square$

Lemma 43. *The following pairs of rules are RE-equivalent:*

(1) $(p; H \leftarrow B, \sim p.)$ *and* $(H \leftarrow B, \sim p.)$;

(2) $(H; \sim p \leftarrow p, B.)$ *and* $(H \leftarrow p, B.)$.

Proof.

(1) Let the first rule be denoted by π_1, the second by π_2 and take some $(I, J) \in \mathfrak{X}$. We need to show that (I, J) is an RE-model of π_1 if and only if it is an RE-model of π_2. Thus, it suffices to prove the following:

$$I \models \pi_1^+ \vee J \models \pi_1^- \qquad \text{if and only if} \qquad I \models \pi_2^+ \vee J \models \pi_2^- \ . \qquad\qquad \text{(A.1)}$$

First note that $\pi_1^- = \pi_2^-$, so $J \models \pi_1^-$ holds if and only if $J \models \pi_2^-$. So if $J \models \pi_1^-$, then $J \models \pi_2^-$ and we can conclude that (A.1) holds. On the other hand, if $J \not\models \pi_1^-$, then $J \not\models \pi_2^-$ and (A.1) reduces to proving that $I \models \pi_1^+$ holds if and only if $I \models \pi_2^+$. Now it suffices to observe that $J \not\models \pi_1^-$ implies $J \models \sim p$ and since I is a subset of J, we can conclude that $I \not\models p$. Since π_1^+ differs from π_2^+ only in the head atom p, our claim follows.

(2) Let the first rule be denoted by π_1, the second by π_2 and take some $(I, J) \in \mathfrak{X}$. We need to show that (I, J) is an RE-model of π_1 if and only if it is an RE-model of π_2. Thus, it suffices to prove the following:

$$I \models \pi_1^+ \vee J \models \pi_1^- \qquad \text{if and only if} \qquad I \models \pi_2^+ \vee J \models \pi_2^- \ . \qquad\qquad \text{(A.2)}$$

First note that $\pi_1^+ = \pi_2^+$, so $I \models \pi_1^+$ holds if and only if $I \models \pi_2^+$. So if $I \models \pi_1^+$, then $I \models \pi_2^+$ and we can conclude that (A.2) holds. On the other hand, if $I \not\models \pi_1^+$, then $I \not\models \pi_2^+$ and (A.2) reduces to proving that $J \models \pi_1^-$ holds if and only if $J \models \pi_2^-$. Now it suffices to observe that $I \not\models \pi_1^+$ implies $I \models p$ and since I is a subset of J, we can conclude that $J \not\models \sim p$. Since π_1^- differs from π_2^- only in the head literal $\sim p$, our claim follows. $\square$

Definition 44 (RE-Canonical Rule)**.** We say that a rule π is *RE-canonical* if either it is τ, or the sets $H_\pi^+ \cup H_\pi^-$, B_π^+ and B_π^- are pairwise disjoint.

Definition 45 (Transformation into an RE-Canonical Rule)**.** Given a rule π, we define the RE-canonical rule $\mathsf{can}_{\mathsf{RE}}(\pi)$ as follows:

(i) If any of the sets $H_\pi^+ \cap B_\pi^+$, $H_\pi^- \cap B_\pi^-$ and $B_\pi^+ \cap B_\pi^-$ is non-empty, then $\mathsf{can}_{\mathsf{RE}}(\pi) = \tau$.

(ii) If (i) does not apply, then $\mathsf{can}_{\mathsf{RE}}(\pi)$ is the rule $\left((H_\pi^+ \setminus B_\pi^-); \overline{(H_\pi^- \setminus B_\pi^+)} \leftarrow B_\pi^+, \overline{B_\pi^-}. \right)$.

Proposition 46. *For every rule π, $[\![\pi]\!]_{\mathsf{RE}} = [\![\mathsf{can}_{\mathsf{RE}}(\pi)]\!]_{\mathsf{RE}}$.*

Proof. This can be shown by a careful iterative application of Lemmas 42 and 43. First observe that if $\mathsf{can}_{\mathsf{RE}}(\pi) = \tau$, then Lemma 42 implies that π is RE-tautological, thus indeed RE-equivalent to τ.

In the principal case we can use Lemma 43(1) on all atoms from $H_\pi^+ \cap B_\pi^-$ and remove them one by one from H_π^+ while preserving RE-models. A similar situation occurs with atoms from $H_\pi^- \cap B_\pi^+$ which can be, according to Lemma 43(2), removed from H_π^- without affecting RE-models. After these steps are performed, the resulting rule coincides with $\mathsf{can}_{\mathsf{RE}}(\pi)$. $\square$

Lemma 47. *For any rule π and $(I, J) \in \mathfrak{X}$, $(I, J) \notin [\![\pi]\!]_{\mathsf{RE}}$ if and only if $B_\pi^+ \subseteq I \subseteq \mathcal{A} \setminus H_\pi^+$ and $H_\pi^- \subseteq J \subseteq \mathcal{A} \setminus B_\pi^-$.*

Proof. Note that $(I, J) \notin [\![\pi]\!]_{\mathsf{RE}}$ if and only if both $I \not\models \pi^+$ and $J \not\models \pi^-$. It can be easily verified that the former is equivalent to $B_\pi^+ \subseteq I \subseteq \mathcal{A} \setminus H_\pi^+$ and the latter to $H_\pi^- \subseteq J \subseteq \mathcal{A} \setminus B_\pi^-$. $\square$

Corollary 48. *Let π be an RE-canonical rule different from τ, put $I = B_\pi^+$, $J = H_\pi^- \cup B_\pi^+$ and $J' = \mathcal{A} \setminus B_\pi^-$, and let p be an atom. Then the following holds:*

(1) $(I, J \cup \{p\}) \in [\![\pi]\!]_{\mathsf{RE}}$ if and only if $p \in B_\pi^-$.

(2) $(I \cup \{p\}, J \cup \{p\}) \in [\![\pi]\!]_{\mathsf{RE}}$ if and only if $p \in H_\pi^+ \cup B_\pi^-$.

(3) $(I \setminus \{p\}, J') \in [\![\pi]\!]_{\mathsf{RE}}$ if and only if $p \in B_\pi^+$.

(4) $(I, J) \notin [\![\pi]\!]_{\mathsf{RE}}$.

Proof. Follows from Lemma 47 and the disjointness properties satisfied by RE-canonical rules. $\square$

Lemma 49. *Let π be an RE-canonical rule different from τ and p an atom. Then the following holds:*

(1) $p \in B_\pi^-$ if and only if for all $(I, J) \in \mathfrak{X}$, $p \in J$ implies $(I, J) \in [\![\pi]\!]_{\mathsf{RE}}$;

(2) $p \in H_\pi^+$ if and only if $p \notin B_\pi^-$ and for all $(I, J) \in \mathfrak{X}$, $p \in I$ implies $(I, J) \in [\![\pi]\!]_{\mathsf{RE}}$;

(3) $p \in B_\pi^+$ if and only if for all $(I, J) \in \mathfrak{X}$, $p \notin I$ implies $(I, J) \in [\![\pi]\!]_{\mathsf{RE}}$;

(4) $p \in H_\pi^-$ if and only if $p \notin B_\pi^+$ and for all $(I, J) \in \mathfrak{X}$, $p \notin J$ implies $(I, J) \in [\![\pi]\!]_{\mathsf{RE}}$.

Proof.

(1) Suppose that $p \in B_\pi^-$ and take some $(I, J) \in \mathfrak{X}$ with $p \in J$. Then $J \not\models \sim p$, so it follows that $J \models \pi^-$. Consequently $(I, J) \in [\![\pi]\!]_{\mathsf{RE}}$.

To prove the converse implication, let $I = B_\pi^+$ and $J = H_\pi^- \cup B_\pi^+$. It follows that $(I, J \cup \{p\}) \in [\![\pi]\!]_{\mathsf{RE}}$, so by Corollary 48(1) we conclude that $p \in B_\pi^-$.

(2) Suppose that $p \in \mathsf{H}_\pi^+$ and take some $(I, J) \in \mathfrak{X}$ with $p \in I$. Then $I \models p$, so it follows that $I \models \pi^+$. Consequently $(I, J) \in [\![\pi]\!]_\mathsf{RE}$.
To prove the converse implication, let $I = \mathsf{B}_\pi^+$ and $J = \mathsf{H}_\pi^- \cup \mathsf{B}_\pi^+$. It follows that $(I \cup \{p\}, J \cup \{p\}) \in [\![\pi]\!]_\mathsf{RE}$, so by Corollary 48(2) we conclude that p belongs to $\mathsf{H}_\pi^+ \cup \mathsf{B}_\pi^-$. Moreover, by the assumption we know that $p \notin \mathsf{B}_\pi^-$, so $p \in \mathsf{H}_\pi^+$.

(3) Suppose that $p \in \mathsf{B}_\pi^+$ and take some $(I, J) \in \mathfrak{X}$ with $p \notin I$. Then $I \not\models p$, so it follows that $I \models \pi^+$. Consequently, $(I, J) \in [\![\pi]\!]_\mathsf{RE}$.
To prove the converse implication, let $I = \mathsf{B}_\pi^+$ and $J' = \mathcal{A} \setminus \mathsf{B}_\pi^-$. It follows that $(I \setminus \{p\}, J') \in [\![\pi]\!]_\mathsf{RE}$, so by Corollary 48(3) we conclude that $p \in \mathsf{B}_\pi^+$.

(4) Suppose that $p \in \mathsf{H}_\pi^-$ and take some $(I, J) \in \mathfrak{X}$ with $p \notin J$. Then $J \models {\sim} p$, so it follows that $J \models \pi^-$. Consequently, $(I, J) \in [\![\pi]\!]_\mathsf{RE}$.
To prove the converse implication, let $I = \mathsf{B}_\pi^+$ and $J = \mathsf{H}_\pi^- \cup \mathsf{B}_\pi^+$. Corollary 48(4) guarantees that $(I, J) \notin [\![\pi]\!]_\mathsf{RE}$. Furthermore, by the assumption it follows that $(I \setminus \{p\}, J \setminus \{p\}) \in [\![\pi]\!]_\mathsf{RE}$. Consequently, J must differ from $J \setminus \{p\}$, which implies that $p \in J$. Furthermore, since $J = \mathsf{H}_\pi^- \cup \mathsf{B}_\pi^+$ and $p \notin \mathsf{B}_\pi^+$ by assumption, we conclude that $p \in \mathsf{H}_\pi^-$. $\qquad\square$

Definition 50 (Rule RE-Induced by a Set of Interpretations)**.** Let $\mathcal{M} \subseteq \mathfrak{X}$. The *rule RE-induced by* $\mathcal{M}$, denoted by $\|\mathcal{M}\|_\mathsf{RE}$, is defined as follows: If $\mathcal{M} = \mathfrak{X}$, then $\|\mathcal{M}\|_\mathsf{RE} = \tau$; otherwise, $\|\mathcal{M}\|_\mathsf{RE}$ is of the form $(H_\mathsf{RE}(\mathcal{M})^+; \overline{H_\mathsf{RE}(\mathcal{M})^-} \leftarrow B_\mathsf{RE}(\mathcal{M})^+, \overline{B_\mathsf{RE}(\mathcal{M})^-}.)$ where

$$
\begin{aligned}
B_\mathsf{RE}(\mathcal{M})^- &= \{\, p \in \mathcal{A} \mid \forall (I, J) \in \mathfrak{X} : p \in J \Rightarrow (I, J) \in \mathcal{M} \,\} \ , \\
H_\mathsf{RE}(\mathcal{M})^+ &= \{\, p \in \mathcal{A} \mid \forall (I, J) \in \mathfrak{X} : p \in I \Rightarrow (I, J) \in \mathcal{M} \,\} \setminus B_\mathsf{RE}(\mathcal{M})^- \ , \\
B_\mathsf{RE}(\mathcal{M})^+ &= \{\, p \in \mathcal{A} \mid \forall (I, J) \in \mathfrak{X} : p \notin I \Rightarrow (I, J) \in \mathcal{M} \,\} \ , \\
H_\mathsf{RE}(\mathcal{M})^- &= \{\, p \in \mathcal{A} \mid \forall (I, J) \in \mathfrak{X} : p \notin J \Rightarrow (I, J) \in \mathcal{M} \,\} \setminus B_\mathsf{RE}(\mathcal{M})^+ \ .
\end{aligned}
$$

Proposition 51. *For every RE-canonical rule π, $\|[\![\pi]\!]_\mathsf{RE}\|_\mathsf{RE} = \pi$.*

Proof. If $\pi = \tau$, then $[\![\pi]\!]_\mathsf{RE} = \mathfrak{X}$ and, by Definition 50, $\|\mathfrak{X}\|_\mathsf{RE} = \tau = \pi$, so the identity is satisfied. In the principal case, π is an RE-canonical rule different from τ. Let $\mathcal{M} = [\![\pi]\!]_\mathsf{RE}$. It follows from Definition 50 and Lemma 49 that $\pi = \|\mathcal{M}\|_\mathsf{RE}$. $\qquad\square$

Corollary 52. *No two different RE-canonical rules are RE-equivalent.*

Proof. Follows directly from Proposition 51. $\qquad\square$

Proof of Proposition 8. Follows from Corollary 52 since every abolishing rule and every constraint is RE-canonical. $\qquad\square$

Lemma 53. *For every rule π, $\mathsf{can}_\mathsf{SE}(\pi) = \mathsf{can}_\mathsf{SE}(\mathsf{can}_\mathsf{RE}(\pi))$.*

Proof. Follows directly from Definitions 38 and 45. $\qquad\square$

Lemma 54. *If π is not RE-equivalent to any abolishing rule, then $\mathsf{can}_\mathsf{RE}(\pi) = \mathsf{can}_\mathsf{SE}(\pi)$.*

Proof. Follows directly from Definitions 38 and 45. $\qquad\square$

Proof of Proposition 9. Suppose that π and σ are RE-equivalent. Then $\mathsf{can}_{\mathsf{RE}}(\pi) = \mathsf{can}_{\mathsf{RE}}(\sigma)$ by Proposition 46 and Corollary 52. By Lemma 53 it follows that $\mathsf{can}_{\mathsf{SE}}(\pi) = \mathsf{can}_{\mathsf{SE}}(\mathsf{can}_{\mathsf{RE}}(\pi)) = \mathsf{can}_{\mathsf{SE}}(\mathsf{can}_{\mathsf{RE}}(\sigma)) = \mathsf{can}_{\mathsf{SE}}(\sigma)$, so by Proposition 39 we can conclude that π, σ are SE-equivalent.

Now suppose that neither π nor σ is RE-equivalent to an abolishing rule and π is RE-equivalent to σ. Then, by Proposition 46, $\mathsf{can}_{\mathsf{RE}}(\pi)$ is RE-equivalent to $\mathsf{can}_{\mathsf{RE}}(\sigma)$ and by Corollary 52, $\mathsf{can}_{\mathsf{RE}}(\pi) = \mathsf{can}_{\mathsf{RE}}(\sigma)$. It follows from Lemma 54 that $\mathsf{can}_{\mathsf{SE}}(\pi) = \mathsf{can}_{\mathsf{SE}}(\sigma)$ and by Proposition 39, π is SE-equivalent to σ.

Finally, by Corollary 52, a rule π is RE-tautological if and only if $\mathsf{can}_{\mathsf{RE}}(\pi) = \tau$ which holds if and only if $\mathsf{can}_{\mathsf{SE}}(\pi) = \tau$ which, by Corollary 40, holds if and only if π is SE-tautological. $\qquad\square$

Proof of Proposition 10. Suppose that J is a stable model of $\mathcal{P}$. Then J is a subset-minimal model of $\mathcal{P}^J$. Thus, (J, J) is an RE-model of $\mathcal{P}$. Now suppose that (I, J) is an RE-model of $\mathcal{P}$ for some $I \subseteq J$. Then $I \models \mathcal{P}^J$ and by the minimality of J we obtain $I = J$.

Suppose that (J, J) is an RE-model of $\mathcal{P}$ and for all $I \subsetneq J$, (I, J) is not an RE-model of $\mathcal{P}$. Then $J \models \mathcal{P}^J$ and J is also a subset-minimal model of $\mathcal{P}^J$. Consequently, J is a stable model of $\mathcal{P}$. $\qquad\square$

Proof of Proposition 11. Let $\mathcal{P}$ contain the rule $\pi_{(I,J)} = \left((\mathcal{A} \setminus I); \overline{J} \leftarrow I, \overline{(\mathcal{A} \setminus J)}. \right)$ for each $(I, J) \in \mathcal{X} \setminus \mathcal{M}$. It is an immediate consequence of Lemma 47 that $[\![\pi_{(I,J)}]\!]_{\mathsf{RE}} = \mathcal{X} \setminus \{(I, J)\}$. Thus,

$$[\![\mathcal{P}]\!]_{\mathsf{RE}} = \bigcap_{(I,J) \in \mathcal{X} \setminus \mathcal{M}} \mathcal{X} \setminus \{(I, J)\} = \mathcal{X} \setminus \bigcup_{(I,J) \in \mathcal{X} \setminus \mathcal{M}} \{(I, J)\} = \mathcal{X} \setminus (\mathcal{X} \setminus \mathcal{M}) = \mathcal{M} \ . \qquad\square$$

Definition 55. A *program entailment relation* is a preorder on the set of all programs. A *program equivalence relation* is an equivalence relation on the set of all programs.

Given a program entailment relation $\models$ and a program equivalence relation $\equiv$, we say that $\models$ *is associated with* $\equiv$ if for all programs $\mathcal{P}, \mathcal{Q}$, $\mathcal{P} \equiv \mathcal{Q}$ holds if and only if $\mathcal{P} \models \mathcal{Q}$ and $\mathcal{Q} \models \mathcal{P}$.

Proof of Proposition 14. If X is SE, RE or SU, then the property follows immediately from the definitions of $\models_{\mathsf{X}}$ and $\equiv_{\mathsf{X}}$.

If X is either SR or RR, then it follows from the definition of $\models_{\mathsf{X}}$ that $\mathcal{P} \models_{\mathsf{X}} \mathcal{Q}$ is equivalent to $\langle\!\langle \mathcal{P}^\tau \rangle\!\rangle_{\mathsf{X}} \supseteq \langle\!\langle \mathcal{Q}^\tau \rangle\!\rangle_{\mathsf{X}}$. Thus, $\mathcal{P} \models_{\mathsf{X}} \mathcal{Q}$ together with $\mathcal{Q} \models_{\mathsf{X}} \mathcal{P}$ is equivalent to $\langle\!\langle \mathcal{P}^\tau \rangle\!\rangle_{\mathsf{X}} = \langle\!\langle \mathcal{Q}^\tau \rangle\!\rangle_{\mathsf{X}}$, which is the definition of $\mathcal{P} \equiv_{\mathsf{X}} \mathcal{Q}$.

It remains to consider the case when X is SMR or RMR. Let Y be SE or RE, respectively. First suppose that $\mathcal{P} \equiv_{\mathsf{X}} \mathcal{Q}$. By the definition of $\equiv_{\mathsf{X}}$ we obtain that $\min\langle\!\langle \mathcal{P}^\tau \rangle\!\rangle_{\mathsf{Y}} = \min\langle\!\langle \mathcal{Q}^\tau \rangle\!\rangle_{\mathsf{Y}}$. Our goal is to prove that $\mathcal{P} \models_{\mathsf{X}} \mathcal{Q}$ and $\mathcal{Q} \models_{\mathsf{X}} \mathcal{P}$. We only show the former; the proof of the latter is analogous. Take some $\sigma \in \mathcal{Q}$. Our goal is find some $\pi \in \mathcal{P}^\tau$ such that $[\![\pi]\!]_{\mathsf{Y}} \subseteq [\![\sigma]\!]_{\mathsf{Y}}$. Take some subset-minimal $\mathcal{M} \in \langle\!\langle \mathcal{Q}^\tau \rangle\!\rangle_{\mathsf{Y}}$ such that $\mathcal{M} \subseteq [\![\sigma]\!]_{\mathsf{Y}}$. It follows from our assumption that $\mathcal{M}$ belongs to $\langle\!\langle \mathcal{P}^\tau \rangle\!\rangle_{\mathsf{Y}}$. In other words, there exists some $\pi \in \mathcal{P}^\tau$ such that $[\![\pi]\!]_{\mathsf{Y}} = \mathcal{M} \subseteq [\![\sigma]\!]_{\mathsf{Y}}$.

Now suppose that both $\mathcal{P} \models_{\mathsf{X}} \mathcal{Q}$ and $\mathcal{Q} \models_{\mathsf{X}} \mathcal{P}$. We need to prove that $\mathcal{P} \equiv_{\mathsf{X}} \mathcal{Q}$, i.e. that $\min\langle\!\langle \mathcal{P}^\tau \rangle\!\rangle_{\mathsf{Y}} = \min\langle\!\langle \mathcal{Q}^\tau \rangle\!\rangle_{\mathsf{Y}}$. We only show that $\min\langle\!\langle \mathcal{P}^\tau \rangle\!\rangle_{\mathsf{Y}} \subseteq \min\langle\!\langle \mathcal{Q}^\tau \rangle\!\rangle_{\mathsf{Y}}$; the proof of the other inclusion is analogical. Take some $\pi \in \mathcal{P}^\tau$ such that

$$[\![\pi]\!]_{\mathsf{Y}} \in \min\langle\!\langle \mathcal{P}^\tau \rangle\!\rangle_{\mathsf{Y}} \ . \tag{A.3}$$

Since $\mathcal{Q} \models_{\mathsf{X}} \mathcal{P}$, there exists some $\sigma \in \mathcal{Q}^\tau$ such that $[\![\sigma]\!]_{\mathsf{Y}} \subseteq [\![\pi]\!]_{\mathsf{Y}}$ Let $\sigma' \in \mathcal{Q}^\tau$ be such that $[\![\sigma']\!]_{\mathsf{Y}} \in \min\langle\!\langle \mathcal{Q}^\tau \rangle\!\rangle_{\mathsf{Y}}$ and $[\![\sigma']\!]_{\mathsf{Y}} \subseteq [\![\sigma]\!]_{\mathsf{Y}}$. Since $\mathcal{P} \models_{\mathsf{X}} \mathcal{Q}$, there exists some $\pi' \in \mathcal{P}^\tau$ such that $[\![\pi']\!]_{\mathsf{Y}} \subseteq [\![\sigma']\!]_{\mathsf{Y}}$. In other words, we know that $[\![\pi']\!]_{\mathsf{Y}} \subseteq [\![\sigma']\!]_{\mathsf{Y}} \subseteq [\![\sigma]\!]_{\mathsf{Y}} \subseteq [\![\pi]\!]_{\mathsf{Y}}$. Thus, due to (A.3) we can conclude that $[\![\pi']\!]_{\mathsf{Y}} = [\![\sigma']\!]_{\mathsf{Y}} = [\![\sigma]\!]_{\mathsf{Y}} = [\![\pi]\!]_{\mathsf{Y}}$, and it follows from the choice of σ' that $[\![\pi]\!]_{\mathsf{Y}} \in \min\langle\!\langle \mathcal{Q}^\tau \rangle\!\rangle_{\mathsf{Y}}$. $\qquad\square$

Lemma 56. *Let $\models_X$, $\models_Y$ be program entailment relations and $\equiv_X$, $\equiv_Y$ program equivalence relations such that $\models_X$ is associated with $\equiv_X$ and $\models_Y$ is associated with $\equiv_Y$. Then $\models_X \preceq \models_Y$ implies $\equiv_X \preceq \equiv_Y$.*

Proof. Suppose that $\models_X \preceq \models_Y$ and take some programs $\mathcal{P}$, $\mathcal{Q}$ such that $\mathcal{P} \equiv_Y \mathcal{Q}$. We need to show that $\mathcal{P} \equiv_X \mathcal{Q}$. Since $\models_Y$ is associated with $\equiv_Y$, we can conclude that $\mathcal{P} \models_Y \mathcal{Q}$ and $\mathcal{Q} \models_Y \mathcal{P}$. Furthermore, from $\models_X \preceq \models_Y$ it follows that $\mathcal{P} \models_X \mathcal{Q}$ and $\mathcal{Q} \models_X \mathcal{P}$, and the assumption that $\models_X$ is associated with $\equiv_X$ implies $\mathcal{P} \equiv_X \mathcal{Q}$. $\qquad\square$

Corollary 57. *Let $\models_X$, $\models_Y$ be program entailment relations and $\equiv_X$, $\equiv_Y$ program equivalence relations such that $\models_X$ is associated with $\equiv_X$ and $\models_Y$ is associated with $\equiv_Y$. Then:*

$$\equiv_X \prec \equiv_Y \ \text{ and } \ \models_X \prec \models_Y \qquad \text{if and only if} \qquad \models_X \preceq \models_Y \ \text{ and } \ \equiv_Y \npreceq \equiv_X \ .$$

Proof. By the definition, $\equiv_X \prec \equiv_Y$ and $\models_X \prec \models_Y$ hold if and only if $\equiv_X \preceq \equiv_Y$, $\equiv_Y \npreceq \equiv_X$, $\models_X \preceq \models_Y$ and $\models_Y \npreceq \models_X$. By Lemma 56, $\models_X \preceq \models_Y$ implies $\equiv_X \preceq \equiv_Y$ and $\equiv_Y \npreceq \equiv_X$ implies $\models_Y \npreceq \models_X$, so the condition can be simplified as desired. $\qquad\square$

Lemma 58. *Let π be a rule and J an interpretation. Then:*

$$J \models \pi \qquad \text{if and only if} \qquad J \models \pi^J \qquad \text{if and only if} \qquad (J, J) \in [\![\pi]\!]_{\mathsf{RE}} \ .$$

Proof. Follows by the definition of π^J and of RE-models. $\qquad\square$

Lemma 59. *Let Π, Σ be rules or programs. Then $[\![\Pi]\!]_{\mathsf{RE}} \subseteq [\![\Sigma]\!]_{\mathsf{RE}}$ implies $[\![\Pi]\!]_{\mathsf{SE}} \subseteq [\![\Sigma]\!]_{\mathsf{SE}}$.*

Proof. Assume that $[\![\Pi]\!]_{\mathsf{RE}} \subseteq [\![\Sigma]\!]_{\mathsf{RE}}$. Then for all $(I, J) \in \mathcal{X}$, $I \models \Pi^J$ implies $I \models \Sigma^J$ and together with Lemma 58 this implies that for all interpretations J,

$$J \models \Pi \qquad \text{implies} \qquad J \models \Pi^J \qquad \text{implies} \qquad J \models \Sigma^J \qquad \text{implies} \qquad J \models \Sigma \ . \qquad (A.4)$$

In order to show that $[\![\Pi]\!]_{\mathsf{SE}} \subseteq [\![\Sigma]\!]_{\mathsf{SE}}$, take some $(I, J) \in [\![\Pi]\!]_{\mathsf{SE}}$. By the definition of SE-models, $J \models \Pi$ and $I \models \Pi^J$, so by (A.4) and our assumption we can conclude that $J \models \Sigma$ and $I \models \Sigma^J$. Thus, by the definition of SE-models, $(I, J) \in [\![\Sigma]\!]_{\mathsf{SE}}$. $\qquad\square$

Proof of Proposition 15.

(1) First we show that $\equiv_{\mathsf{SM}} \prec \equiv_{\mathsf{SE}}$, i.e. that $\equiv_{\mathsf{SM}} \preceq \equiv_{\mathsf{SE}}$ and $\equiv_{\mathsf{SE}} \npreceq \equiv_{\mathsf{SM}}$. To verify the former, suppose that $\mathcal{P}$, $\mathcal{Q}$ are programs with $\mathcal{P} \equiv_{\mathsf{SE}} \mathcal{Q}$. Then $\mathcal{P} \cup \emptyset$ has the same stable models as $\mathcal{Q} \cup \emptyset$. Consequently, $\mathcal{P} \equiv_{\mathsf{SM}} \mathcal{Q}$. To see that $\equiv_{\mathsf{SE}} \npreceq \equiv_{\mathsf{SM}}$, observe that the programs $\mathcal{P} = \emptyset$, $\mathcal{Q} = \{\, p \leftarrow q. \,\}$ are SM-equivalent but not SE-equivalent.

Turning to the remaining relationships, it follows from Corollary 57 that we can instead prove that

$$\models_{\mathsf{SE}} \preceq \models_{\mathsf{RE}} \preceq \models_{\mathsf{RMR}} \preceq \models_{\mathsf{RR}} \preceq \models_{\mathsf{SU}} \qquad \text{and} \qquad \equiv_{\mathsf{SU}} \npreceq \equiv_{\mathsf{RR}} \npreceq \equiv_{\mathsf{RMR}} \npreceq \equiv_{\mathsf{RE}} \npreceq \equiv_{\mathsf{SE}} \ . \qquad (A.5)$$

We first concentrate on the left-hand side of (A.5). In order to show that $\models_{\mathsf{SE}} \preceq \models_{\mathsf{RE}}$, suppose that $\mathcal{P}$, $\mathcal{Q}$ are programs such that $\mathcal{P} \models_{\mathsf{RE}} \mathcal{Q}$. Then $[\![\mathcal{P}]\!]_{\mathsf{RE}} \subseteq [\![\mathcal{Q}]\!]_{\mathsf{RE}}$ and it follows from Lemma 59 that $[\![\mathcal{P}]\!]_{\mathsf{SE}} \subseteq [\![\mathcal{Q}]\!]_{\mathsf{SE}}$. Consequently, $\mathcal{P} \models_{\mathsf{SE}} \mathcal{Q}$.

We also need to prove that $\models_{\mathsf{RE}} \preceq \models_{\mathsf{RMR}}$. Take some programs $\mathcal{P}$, $\mathcal{Q}$ with $\mathcal{P} \models_{\mathsf{RMR}} \mathcal{Q}$ and put $\mathcal{P}^\tau = \mathcal{P} \cup \{\, \tau \,\}$. It follows that

$$\forall \sigma \in \mathcal{Q} \, \exists \pi_\sigma \in \mathcal{P}^\tau : [\![\pi_\sigma]\!]_{\mathsf{RE}} \subseteq [\![\sigma]\!]_{\mathsf{RE}} \ . \qquad (A.6)$$

We need to prove that $[\![\mathcal{P}]\!]_{\mathsf{RE}} \subseteq [\![\mathcal{Q}]\!]_{\mathsf{RE}}$. Suppose that $X \in [\![\mathcal{P}]\!]_{\mathsf{RE}}$ and take some $\sigma \in \mathcal{Q}$, our goal is to show that $X \in [\![\sigma]\!]_{\mathsf{RE}}$. By (A.6) there exists some $\pi_\sigma \in \mathcal{P}^\tau$ such that $[\![\pi_\sigma]\!]_{\mathsf{RE}} \subseteq [\![\sigma]\!]_{\mathsf{RE}}$. If $\pi_\sigma = \tau$, then it immediately follows that $X \in \mathcal{X} = [\![\tau]\!]_{\mathsf{RE}} \subseteq [\![\sigma]\!]_{\mathsf{RE}}$. If $\pi_\sigma \in \mathcal{P}$, then $X \in [\![\pi_\sigma]\!]_{\mathsf{RE}}$ by the choice of X, so $X \in [\![\sigma]\!]_{\mathsf{RE}}$.

Our next goal is to show that $\models_{\mathsf{RMR}} \preceq \models_{\mathsf{RR}}$. This follows directly by the definitions of $\models_{\mathsf{RMR}}$ and $\models_{\mathsf{RR}}$. To prove the final part of the left-hand side of (A.5), suppose that $\mathcal{P} \models_{\mathsf{SU}} \mathcal{Q}$. Then $[\![\mathcal{Q} \setminus \mathcal{P}]\!]_{\mathsf{SE}} = \mathcal{X}$. We need to prove that $\mathcal{P} \models_{\mathsf{RR}} \mathcal{Q}$, i.e. that for every $\sigma \in \mathcal{Q}$ there is some $\pi \in \mathcal{P}^\tau$ such that $[\![\pi]\!]_{\mathsf{RE}} = [\![\sigma]\!]_{\mathsf{RE}}$. Pick some $\sigma \in \mathcal{Q}$ and note that $\mathcal{Q} = (\mathcal{Q} \cap \mathcal{P}) \cup (\mathcal{Q} \setminus \mathcal{P})$. If $\sigma \in \mathcal{Q} \cap \mathcal{P}$, then $\sigma \in \mathcal{P}$ and we can put $\pi = \sigma$ to finish the proof. In the remaining case, $\sigma \in \mathcal{Q} \setminus \mathcal{P}$ and it follows from our assumption that $[\![\sigma]\!]_{\mathsf{SE}} = \mathcal{X}$. Thus, putting $\pi = \tau$ finishes the proof.

As for the right-hand side of (A.5), we can see that $\equiv_{\mathsf{SU}} \not\preceq \equiv_{\mathsf{RR}}$ because the programs $\mathcal{P} = \{\, \sim p \leftarrow p. \,\}$ and $\mathcal{Q} = \{\, \leftarrow p. \,\}$ are RR-equivalent but not SU-equivalent.

Similarly, programs $\mathcal{P} = \{\, p. \,\}$ and $\mathcal{Q} = \{\, p., p \leftarrow q. \,\}$ are RMR-equivalent but not RR-equivalent, so it follows that $\equiv_{\mathsf{RR}} \not\preceq \equiv_{\mathsf{RMR}}$.

Next, to verify that $\equiv_{\mathsf{RMR}} \not\preceq \equiv_{\mathsf{RE}}$ it suffices to observe that the programs $\mathcal{P} = \{\, p., q. \,\}$ and $\mathcal{Q} = \{\, p., q \leftarrow p. \,\}$ are RE-equivalent but not RMR-equivalent.

Finally, programs $\mathcal{P} = \{\, \sim p. \,\}$ and $\mathcal{Q} = \{\, \leftarrow p. \,\}$ are SE-equivalent but not RE-equivalent, proving that $\equiv_{\mathsf{RE}} \not\preceq \equiv_{\mathsf{SE}}$.

(2) It follows from Corollary 57 that we can instead prove that

$$\models_{\mathsf{SE}} \preceq \models_{\mathsf{SMR}} \preceq \models_{\mathsf{SR}} \preceq \models_{\mathsf{RR}} \qquad \text{and} \qquad \equiv_{\mathsf{RR}} \not\preceq \equiv_{\mathsf{SR}} \not\preceq \equiv_{\mathsf{SMR}} \not\preceq \equiv_{\mathsf{SE}} \quad . \tag{A.7}$$

We first concentrate on the left-hand side of (A.7). To prove that $\models_{\mathsf{SE}} \preceq \models_{\mathsf{SMR}}$, take some programs $\mathcal{P}, \mathcal{Q}$ with $\mathcal{P} \models_{\mathsf{SMR}} \mathcal{Q}$ and put $\mathcal{P}^\tau = \mathcal{P} \cup \{\, \tau \,\}$. It follows that

$$\forall \sigma \in \mathcal{Q} \, \exists \pi_\sigma \in \mathcal{P}^\tau : [\![\pi_\sigma]\!]_{\mathsf{SE}} \subseteq [\![\sigma]\!]_{\mathsf{SE}} \quad . \tag{A.8}$$

We need to prove that $[\![\mathcal{P}]\!]_{\mathsf{SE}} \subseteq [\![\mathcal{Q}]\!]_{\mathsf{SE}}$. Suppose that $X \in [\![\mathcal{P}]\!]_{\mathsf{SE}}$ and take some $\sigma \in \mathcal{Q}$, our goal is to show that $X \in [\![\sigma]\!]_{\mathsf{SE}}$. By (A.8) there exists some $\pi_\sigma \in \mathcal{P}^\tau$ such that $[\![\pi_\sigma]\!]_{\mathsf{SE}} \subseteq [\![\sigma]\!]_{\mathsf{SE}}$. If $\pi_\sigma = \tau$, then it immediately follows that $X \in \mathcal{X} = [\![\tau]\!]_{\mathsf{SE}} \subseteq [\![\sigma]\!]_{\mathsf{SE}}$. If $\pi_\sigma \in \mathcal{P}$, then $X \in [\![\pi_\sigma]\!]_{\mathsf{SE}}$ by the choice of X, so $X \in [\![\sigma]\!]_{\mathsf{SE}}$.

Our next goal is to show that $\models_{\mathsf{SMR}} \preceq \models_{\mathsf{SR}}$. This follows directly by the definitions of $\models_{\mathsf{SMR}}$ and $\models_{\mathsf{SR}}$. To finish the proof of the left-hand side of (A.5), suppose that $\mathcal{P} \models_{\mathsf{RR}} \mathcal{Q}$. Then $\forall \sigma \in \mathcal{Q} \, \exists \pi \in \mathcal{P} : [\![\pi]\!]_{\mathsf{RE}} = [\![\sigma]\!]_{\mathsf{RE}}$ and, due to Lemma 59, $\forall \sigma \in \mathcal{Q} \, \exists \pi \in \mathcal{P} : [\![\pi]\!]_{\mathsf{SE}} = [\![\sigma]\!]_{\mathsf{SE}}$. Consequently, $\mathcal{P} \models_{\mathsf{SR}} \mathcal{Q}$.

As for the right-hand side of (A.5), we can see that $\equiv_{\mathsf{RR}} \not\preceq \equiv_{\mathsf{SR}}$ because the programs $\mathcal{P} = \{\, \sim p. \,\}$ and $\mathcal{Q} = \{\, \leftarrow p. \,\}$ are SR-equivalent but not RR-equivalent.

Similarly, programs $\mathcal{P} = \{\, p. \,\}$ and $\mathcal{Q} = \{\, p., p \leftarrow q. \,\}$ are SMR-equivalent but not SR-equivalent, so it follows that $\equiv_{\mathsf{SR}} \not\preceq \equiv_{\mathsf{SMR}}$.

Finally, to verify that $\equiv_{\mathsf{SMR}} \not\preceq \equiv_{\mathsf{SE}}$ it suffices to observe that the programs $\mathcal{P} = \{\, p., q. \,\}$ and $\mathcal{Q} = \{\, p., q \leftarrow p. \,\}$ are SE-equivalent but not SMR-equivalent.

(3) It follows from Corollary 57 that we can instead prove that $\models_{\mathsf{SMR}} \preceq \models_{\mathsf{RMR}}$ and $\equiv_{\mathsf{RMR}} \not\preceq \equiv_{\mathsf{SMR}}$. To show the former, take some programs $\mathcal{P}, \mathcal{Q}$ such that $\mathcal{P} \models_{\mathsf{RMR}} \mathcal{Q}$. It follows that $\forall \sigma \in \mathcal{Q} \, \exists \pi \in \mathcal{P} : [\![\pi]\!]_{\mathsf{RE}} \subseteq [\![\sigma]\!]_{\mathsf{RE}}$ and, due to Lemma 59, $\forall \sigma \in \mathcal{Q} \, \exists \pi \in \mathcal{P} : [\![\pi]\!]_{\mathsf{SE}} \subseteq [\![\sigma]\!]_{\mathsf{SE}}$. Consequently, $\mathcal{P} \models_{\mathsf{SMR}} \mathcal{Q}$. As for the latter, it suffices to observe that the programs $\mathcal{P} = \{\, \sim p. \,\}$ and $\mathcal{Q} = \{\, \leftarrow p. \,\}$ are SMR-equivalent but not RMR-equivalent.

(4) According to Lemma 56, it suffices to show that $\equiv_{\mathsf{RE}} \not\preceq \equiv_{\mathsf{SMR}}$ and $\equiv_{\mathsf{SMR}} \not\preceq \equiv_{\mathsf{RE}}$. The former follows from the fact that the programs $\mathcal{P} = \{\, \sim p. \,\}$ and $\mathcal{Q} = \{\, \leftarrow p. \,\}$ are SMR-equivalent but not

RE-equivalent. The latter can be verified by observing that though the programs $\mathcal{P} = \{\, p., q. \,\}$ and $\mathcal{Q} = \{\, p., q \leftarrow p. \,\}$ are RE-equivalent, they are not SMR-equivalent.

(5) According to Lemma 56, it suffices to show that $\equiv_{\mathsf{RE}} \not\leq \equiv_{\mathsf{SR}}$ and $\equiv_{\mathsf{SR}} \not\leq \equiv_{\mathsf{RE}}$. The former follows from the fact that the programs $\mathcal{P} = \{\, \sim p. \,\}$ and $\mathcal{Q} = \{\, \leftarrow p. \,\}$ are SR-equivalent but not RE-equivalent. The latter can be verified by observing that though the programs $\mathcal{P} = \{\, p., q. \,\}$ and $\mathcal{Q} = \{\, p., q \leftarrow p. \,\}$ are RE-equivalent, they are not SR-equivalent.

(6) According to Lemma 56, it suffices to show that $\equiv_{\mathsf{RMR}} \not\leq \equiv_{\mathsf{SR}}$ and $\equiv_{\mathsf{SR}} \not\leq \equiv_{\mathsf{RMR}}$. The former follows from the fact that the programs $\mathcal{P} = \{\, \sim p. \,\}$ and $\mathcal{Q} = \{\, \leftarrow p. \,\}$ are SR-equivalent but not RMR-equivalent. The latter can be verified by observing that though the programs $\mathcal{P} = \{\, p. \,\}$ and $\mathcal{Q} = \{\, p., p \leftarrow q. \,\}$ are RMR-equivalent, they are not SR-equivalent. $\qquad\square$

Appendix B. Proofs: Exception-Based Rule Update Operators

Appendix B.1. Conflicts Between Sets of RE-Models

Definition 60 (Additional Notation). Let $X \in \mathcal{X}$. Given an atom p, we say that X is an *RE-model of p* if $X(p) = \mathsf{T}$. We say that X is an *RE-model of $\sim p$* if $X(p) = \mathsf{F}$. We denote the set of all RE-models of a literal L by $[\![L]\!]_{\mathsf{RE}}$. Given a set of literals S, we say that X is an *RE-model of S* if X is an RE-model of all literals in S. We denote the set of all RE-models of S by $[\![S]\!]_{\mathsf{RE}}$. Given a sequence of rule bases $\boldsymbol{R} = \langle \mathcal{R}_i \rangle_{i<n}$, we define $\langle\!\langle \boldsymbol{R} \rangle\!\rangle_{\mathsf{RE}} = \langle \langle\!\langle \mathcal{R}_i \rangle\!\rangle_{\mathsf{RE}} \rangle_{i<n}$.

Proposition 61. *Let π be a rule, p an atom and $X = (I, J) \in \mathcal{X}$. Then $(I \setminus \{\, p \,\}, J \cup \{\, p \,\})$ is not an RE-model of π if and only if the following holds:*

1. *Neither p nor $\sim p$ belongs to B_π;*

2. *X is an RE-model of B_π;*

3. *X is not an RE-model of any literal from $\mathsf{H}_\pi \setminus \{\, p, \sim p \,\}$.*

Proof. We prove the direct implication first. From the assumption it follows that $\pi^{J \cup \{ p \}}$ is different from τ. This has two consequences. First, H_π^- is included in $J \cup \{\, p \,\}$, so all atoms from H_π^-, except possibly p, belong to J, and thus X is not an RE-model of any default literal from $\mathsf{H}_\pi \setminus \{\, \sim p \,\}$. The second consequence is that $\mathsf{B}_\pi^- \cap (J \cup \{\, p \,\})$ is empty. Hence, $\sim p$ does not belong to B_π. Furthermore, $\mathsf{B}_\pi^- \cap J$ must also be empty, so we can conclude that X is an RE-model of all default literals from B_π. It also follows from the assumption that $I \setminus \{\, p \,\}$ contains B_π^+ but does not contain any atom from H_π^+. As a consequence, p does not belong to B_π and we can also conclude that X is an RE-model of all atoms from B_π and X is not an RE-model of any atom from $\mathsf{H}_\pi \setminus \{\, p \,\}$. Thus, X is an RE-model of B_π and it is not an RE-model of any literal from $\mathsf{H}_\pi \setminus \{\, p, \sim p \,\}$.

As for the converse implication, we need to prove that $I \setminus \{\, p \,\}$ is not a model of $\pi^{J \cup \{ p \}}$. We first need to show that $\pi^{J \cup \{ p \}}$ is equal to the rule $\mathsf{H}_\pi^+ \leftarrow \mathsf{B}_\pi^+$. This holds if H_π^- is included in $J \cup \{\, p \,\}$ and B_π^- is disjoint with $J \cup \{\, p \,\}$. Since X is an RE-model of B_π, we can conclude that the set B_π^- is disjoint with J which, together with the assumption that $\sim p$ does not belong to B_π, implies that B_π^- is disjoint with $J \cup \{\, p \,\}$. We also know that X is not an RE-model of any literal from $\mathsf{H}_\pi \setminus \{\, p, \sim p \,\}$, so we can conclude that $\mathsf{H}_\pi^- \setminus \{\, p \,\}$ is included in J. Thus, H_π^- is included in $J \cup \{\, p \,\}$ and we proved that $\pi^{J \cup \{ p \}}$ is equal to the rule $\mathsf{H}_\pi^+ \leftarrow \mathsf{B}_\pi^+$. It remains to show that $I \setminus \{\, p \,\}$ includes B_π^+ and that it does not contain any atom from H_π^+. We know that X is an RE-model of B_π, so I includes B_π^+. Also, since p does not belong to B_π, $I \setminus \{\, p \,\}$ also includes B_π^+. Finally, we know that X is not an RE-model of any atom from $\mathsf{H}_\pi^+ \setminus \{\, p \,\}$, so I does not contain any atom from $\mathsf{H}_\pi^+ \setminus \{\, p \,\}$. This implies that $I \setminus \{\, p \,\}$ does not contain any atom from H_π^+. $\qquad\square$

Corollary 62. *Let π be a rule, p an atom and J an interpretation with $p \in J$. If (J, J) is an RE-model of π but $(J \setminus \{\, p \,\}, J)$ is not, then $p \in \mathsf{H}_\pi$ and $J \models \mathsf{B}_\pi$.*

Proof. It follows immediately from Proposition 61 that $J \models \mathsf{B}_\pi$. Furthermore, by the definition of RE-model, J is a model of π^J while $J \setminus \{\, p \,\}$ is not. Hence, J contains some atom from H_π that is not contained in $J \setminus \{\, p \,\}$. This atom can only be p. $\qquad\square$

Proposition 63. *Let π be a rule, $\mathcal{M} = [\![\pi]\!]_{\mathsf{RE}}$, p an atom, J an interpretation and V a truth value. Then $\mathcal{M}^J(p) = \mathsf{V}$ if and only if the following holds:*

1. *Neither p nor $\sim p$ belongs to B_π;*

2. *J is a model of B_π;*

3. *J is not a model of any literal from $\mathsf{H}_\pi \setminus \{\, p, \sim p \,\}$;*

4. *One of the following conditions holds:*

    (a) *V is T and $\mathsf{H}_\pi \cap \{\, p, \sim p \,\} = \{\, p \,\}$, or*

    (b) *V is F and $\mathsf{H}_\pi \cap \{\, p, \sim p \,\} = \{\, \sim p \,\}$.*

Proof. We focus on the direct implication first. Assume that $\mathsf{V} = \mathsf{T}$. Then $(J \cup \{\, p \,\}, J \cup \{\, p \,\})$ is an RE-model of π although both $(J \setminus \{\, p \,\}, J \cup \{\, p \,\})$ and $(J \setminus \{\, p \,\}, J \setminus \{\, p \,\})$ are not. By Proposition 61 and Lemma 58 we can conclude that the first three of the properties that we need to prove are satisfied. It remains to show that $\mathsf{H}_\pi \cap \{\, p, \sim p \,\} = \{\, p \,\}$, i.e. that p belongs to H_π^+ but it does not belong to H_π^-. To see that the former holds, note that $J \setminus \{\, p \,\}$ is not a model of $\pi^{J \cup \{\, p \,\}}$, so $J \setminus \{\, p \,\}$ includes B_π^+ and it does not contain any atom from H_π^+. Since we know that $J \cup \{\, p \,\}$ *is* a model of $\pi^{J \cup \{\, p \,\}}$, it must be the case that $J \cup \{\, p \,\}$ contains an atom from H_π^+. This atom can only be p. Finally, if p were a member of H_π^-, then $\pi^{J \setminus \{\, p \,\}}$ would coincide with τ, so $(J \setminus \{\, p \,\}, J \setminus \{\, p \,\})$ would be an RE-model of π, contrary to the assumption.

Now assume that $\mathsf{V} = \mathsf{F}$. Then $(J \setminus \{\, p \,\}, J \setminus \{\, p \,\})$ is an RE-model of π although both $(J \setminus \{\, p \,\}, J \cup \{\, p \,\})$ and $(J \cup \{\, p \,\}, J \cup \{\, p \,\})$ are not. By Proposition 61 and Lemma 58 we can conclude that the first three of the properties that we need to prove are satisfied. It remains to show that $\mathsf{H}_\pi \cap \{\, p, \sim p \,\} = \{\, \sim p \,\}$, i.e. that p belongs to H_π^- but it does not belong to H_π^+. To see that the former holds, note that by the assumption $J \setminus \{\, p \,\}$ is a model of $\pi^{J \setminus \{\, p \,\}}$ while it is not a model of $\pi^{J \cup \{\, p \,\}}$. Hence, $\pi^{J \setminus \{\, p \,\}}$ must be equal to τ. We know that B_π^- is disjoint with $J \cup \{\, p \,\}$, so it must also be disjoint with $J \setminus \{\, p \,\}$. Thus, there must exist some atom from H_π^- that is not contained in $J \setminus \{\, p \,\}$ while it was contained in $J \cup \{\, p \,\}$. This atom can only be p. Finally, if p were a member of H_π^+, then $J \cup \{\, p \,\}$ would contain an atom from H_π^+, so $(J \cup \{\, p \,\}, J \cup \{\, p \,\})$ would be an RE-model of π, contrary to the assumption.

Finally, we show by contradiction that V cannot be equal to U. Suppose that $\mathsf{V} = \mathsf{U}$. It follows that $(J \cup \{\, p \,\}) \not\models \pi^{J \cup \{\, p \,\}}$, $(J \setminus \{\, p \,\}) \models \pi^{J \cup \{\, p \,\}}$ and $(J \setminus \{\, p \,\}) \not\models \pi^{J \setminus \{\, p \,\}}$. Thus, since both $\pi^{J \setminus \{\, p \,\}}$ and $\pi^{J \cup \{\, p \,\}}$ are different from τ, they must be identical and this is in conflict with our latter two conclusions.

As for the converse implication, let $X^{\mathsf{T}} = (J \cup \{\, p \,\}, J \cup \{\, p \,\})$, $X^{\mathsf{U}} = (J \setminus \{\, p \,\}, J \cup \{\, p \,\})$ and $X^{\mathsf{F}} = (J \setminus \{\, p \,\}, J \setminus \{\, p \,\})$ First suppose that V is T and $\mathsf{H}_\pi \cap \{\, p, \sim p \,\} = \{\, p \,\}$. We need to show that X^{T} is an RE-model of π while both X^{U} and X^{F} are not. The first property follows directly from the fact that p belongs to H_π^+ and X^{T} is an RE-model of p. The second property follows from Proposition 61 and Lemma 58. To show that the third is also satisfied, note that since X^{U} is not an RE-model of π, the rule $\pi^{J \cup \{\, p \,\}}$ coincides with the rule $\mathsf{H}_\pi^+ \leftarrow \mathsf{B}_\pi^+$. This implies that B_π^- is disjoint with $J \cup \{\, p \,\}$ and H_π^- is included in $J \cup \{\, p \,\}$. As a consequence, B_π^- is also disjoint with $J \setminus \{\, p \,\}$. Moreover, from our assumptions we know that $\mathsf{H}_\pi \cap \{\, p, \sim p \,\} = \{\, p \,\}$, which means that p does not belong to H_π^-. Thus, H_π^- is included in $J \setminus \{\, p \,\}$.

As a consequence, the rule $\pi^{J \setminus \{p\}}$ also coincides with the rule $\mathsf{H}_\pi^+ \leftarrow \mathsf{B}_\pi^+$. Furthermore, since X^U is not an RE-model of π, $J \setminus \{p\}$ is not a model of $\pi^{J \cup \{p\}}$. Since $\pi^{J \cup \{p\}} = \pi^{J \setminus \{p\}}$, we obtain that $J \setminus \{p\}$ is not a model of $\pi^{J \setminus \{p\}}$. Hence X^F is not an RE-model of π.

Next, suppose that V is F and $\mathsf{H}_\pi \cap \{p, \sim p\} = \{\sim p\}$. We need to show that X^F is an RE-model of π while both X^U and X^T are not. The first property follows directly from the fact that p belongs to H_π^- but does not belong to $J \setminus \{p\}$ because in this case $\pi^{J \setminus \{p\}}$ coincides with τ. The second property follows from Proposition 61 and Lemma 58. To show that the third is also satisfied, note that since X^U is not an RE-model of π, the rule $\pi^{J \cup \{p\}}$ coincides with the rule $\mathsf{H}_\pi^+ \leftarrow \mathsf{B}_\pi^+$ and $J \setminus \{p\}$ is not a model of $\pi^{J \cup \{p\}}$, i.e. $J \setminus \{p\}$ includes B_π^+ but does not contain any atom from H_π^+. Thus, $J \cup \{p\}$ also includes B_π^+ and from our assumption that $\mathsf{H}_\pi \cap \{p, \sim p\} = \{\sim p\}$ we can conclude that p does not belong to H_π^+. Thus, $J \cup \{p\}$ does not contain any atom from H_π^+ and, consequently, X^T is not an RE-model of π. $\qquad\square$

Proposition 64. *Let π and σ be non-disjunctive rules, $\mathcal{M} = [\![\pi]\!]_\mathsf{RE}$, $\mathcal{N} = [\![\sigma]\!]_\mathsf{RE}$, and J an interpretation. Then $\mathcal{M} \bowtie_p^J \mathcal{N}$ if and only if for some $L \in \{p, \sim p\}$, $\mathsf{H}_\pi = \{L\}$, $\mathsf{H}_\sigma = \{\overline{L}\}$, J is a model of both B_π and B_σ, and B_π, B_σ do not contain p nor $\sim p$.*

Proof. Follows directly from Proposition 63. $\qquad\square$

Appendix B.2. Syntactic Properties of δ_a-Based Operators

Definition 65. Let δ be a local exception function, $\mathcal{M} \subseteq \mathcal{X}$ and $\mathcal{S} \subseteq 2^\mathcal{X}$. We define $\mathsf{aug}_\delta(\mathcal{M}, \mathcal{S}) = \mathcal{M} \cup \bigcup_{\mathcal{N} \in \mathcal{S}} \delta(\mathcal{M}, \mathcal{N})$ and extend this definition to sequences of sets of sets of three-valued interpretations inductively as follows: $\mathsf{aug}_\delta(\mathcal{M}, \langle\rangle) = \mathcal{M}$ and $\mathsf{aug}_\delta(\mathcal{M}, \langle \mathcal{S}_i \rangle_{i<n+1}) = \mathsf{aug}_\delta(\mathsf{aug}_\delta(\mathcal{M}, \langle \mathcal{S}_i \rangle_{i<n}), \mathcal{S}_n)$.

Proposition 66. *Let $\boldsymbol{R} = \langle \mathcal{R}_i \rangle_{i<n}$ be a sequence of rule bases, $\mathcal{S}_i = \langle\!\langle \mathcal{R}_i \rangle\!\rangle_\mathsf{RE}$ for all $i < n$, and $\oplus$ a δ-based rule update operator. Then, $\langle\!\langle \bigoplus \boldsymbol{R} \rangle\!\rangle_\mathsf{RE} = \{\mathsf{aug}_\delta(\mathcal{M}, \langle \mathcal{S}_j \rangle_{i<j<n}) \mid i < n \wedge \mathcal{M} \in \mathcal{S}_i\}$.*

Proof. Follows by induction on n. $\qquad\square$

Lemma 67. *Let $\mathcal{M} \subseteq \mathcal{X}$, $\mathcal{S} = \langle \mathcal{S}_i \rangle_{i<n}$ where $\mathcal{S}_i \subseteq 2^\mathcal{X}$ for all $i < n$, J an interpretation and p an atom. If (J, J) belongs to $\mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}, \mathcal{S})$, but $(J \setminus \{p\}, J)$ does not, then (J, J) belongs to $\mathcal{M}$.*

Proof. Follows by induction on n. $\qquad\square$

Proposition 68. *Every δ_a-based rule update operator respects support.*

Proof. Let $\oplus$ be some δ_a-based rule update operator, pick some DLP $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$, suppose that J is a stable model of $\bigoplus \boldsymbol{P}$ and take some $p \in J$. We need to show that for some rule $\pi \in \mathsf{all}(\boldsymbol{P})$, $p \in \mathsf{H}_\pi$ and $J \models \mathsf{B}_\pi$. Since J is a stable model of $\bigoplus \boldsymbol{P}$, we know that (J, J) belongs to $[\![\bigoplus \boldsymbol{P}]\!]_\mathsf{RE}$ and for all $I \subsetneq J$, (I, J) does not belong to $[\![\bigoplus \boldsymbol{P}]\!]_\mathsf{RE}$. In particular, $(J \setminus \{p\}, J)$ does not belong to $[\![\bigoplus \boldsymbol{P}]\!]_\mathsf{RE}$. Consequently, there is some $\mathcal{N} \in \langle\!\langle \bigoplus \boldsymbol{P} \rangle\!\rangle_\mathsf{RE}$ such that $(J \setminus \{p\}, J)$ does not belong to $\mathcal{N}$ although (J, J) does. According to Proposition 66, $\mathcal{N} = \mathsf{aug}_\delta(\mathcal{M}, \langle\!\langle \mathcal{P}_j \rangle\!\rangle_\mathsf{RE} \rangle_{i<j<n})$ where $\mathcal{M} \in \langle\!\langle \mathcal{P}_i \rangle\!\rangle_\mathsf{RE}$ for some $i < n$. Let π be a rule from $\mathcal{P}_i$ such that $[\![\pi]\!]_\mathsf{RE} = \mathcal{M}$. Since $(J \setminus \{p\}, J)$ does not belong to $\mathcal{N}$, it cannot belong to $\mathcal{M}$ since $\mathcal{M}$ is a subset of $\mathcal{N}$. Also, by Lemma 67, (J, J) belongs to $\mathcal{M}$. Thus, it follows from Corollary 62 that $p \in \mathsf{H}_\pi$ and $J \models \mathsf{B}_\pi$. $\qquad\square$

Lemma 69. *Let $\boldsymbol{P}$ be a finite sequence of sets of facts and L a literal. Then,*

$$\mathsf{aug}_{\delta_\mathsf{a}}([\![L.]\!]_\mathsf{RE}, \langle\!\langle \boldsymbol{P} \rangle\!\rangle_\mathsf{RE}) = \begin{cases} \mathcal{X} & (\overline{L}.) \in \mathsf{all}(\boldsymbol{P}) \ ; \\ [\![L.]\!]_\mathsf{RE} & otherwise \ . \end{cases}$$

40

Proof. Follows using Proposition 64 by induction on the length of $\boldsymbol{P}$. □

Corollary 70. *Let $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ be a finite sequence of sets of facts and $\oplus$ a δ_{a}-based rule update operator. Then, $\langle\!\langle \bigoplus \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}} \cup \{ \mathfrak{X} \} = \{ [\![L.]\!]_{\mathsf{RE}} \mid \exists i < n : (L.) \in \mathcal{P}_i \wedge (\forall j : i < j < n \Rightarrow (\overline{L}.) \notin \mathcal{P}_j) \} \cup \{ \mathfrak{X} \}$.*

Proof. By Proposition 66, $\langle\!\langle \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}} = \{ \mathsf{aug}_\delta([\![\pi]\!]_{\mathsf{RE}}, \langle \langle\!\langle \mathcal{P}_j \rangle\!\rangle_{\mathsf{RE}} \rangle_{i<j<n}) \mid i < n \wedge \pi \in \mathcal{P}_i \}$, which can also be written as $\{ \mathsf{aug}_\delta([\![L.]\!]_{\mathsf{RE}}, \langle \langle\!\langle \mathcal{P}_j \rangle\!\rangle_{\mathsf{RE}} \rangle_{i<j<n}) \mid i < n \wedge (L.) \in \mathcal{P}_i \}$. Furthermore, due to Lemma 69, we can equivalently write this as

$$\big\{ \mathfrak{X} \mid \exists i, j, L : i < j < n \wedge (L.) \in \mathcal{P}_i \wedge (\overline{L}.) \in \mathcal{P}_j \big\}$$
$$\cup \big\{ [\![L.]\!]_{\mathsf{RE}} \mid \exists i < n : (L.) \in \mathcal{P}_i \wedge (\forall j : i < j < n \Rightarrow (\overline{L}.) \notin \mathcal{P}_j) \big\} \ .$$

This directly implies the desired conclusion. □

Proposition 71. *Every δ_{a}-based rule update operator respects fact update.*

Proof. Let $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ be a finite sequence of consistent sets of facts, J the interpretation

$$\{ p \mid \exists i < n : (p.) \in \mathcal{P}_i \wedge (\forall j : i < j < n \Rightarrow (\sim p.) \notin \mathcal{P}_j) \}$$

and $\oplus$ a δ_{a}-based rule update operator. We need to show that J is the unique stable model of $\bigoplus \boldsymbol{P}$.

We start by proving that (J, J) belongs to $[\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{RE}}$. Pick some $\mathcal{M} \in \langle\!\langle \bigoplus \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}}$. By Corollary 70 we know that $\mathcal{M}$ is either $\mathfrak{X}$, or it is equal to $[\![L.]\!]_{\mathsf{RE}}$ where

$$\exists i < n : (L.) \in \mathcal{P}_i \wedge (\forall j : i < j < n \Rightarrow (\overline{L}.) \notin \mathcal{P}_j) \ .$$

In the former case it trivially holds that (J, J) belongs to $\mathcal{M} = \mathfrak{X}$. Now suppose that L is an atom p. Then, by its definition, J contains p, so (J, J) belongs to $[\![p.]\!]_{\mathsf{RE}} = \mathcal{M}$. On the other hand, if L is a default literal $\sim p$, then the fact $(p.)$ does not belong to $\mathcal{P}_i$ because $\mathcal{P}_i$ is consistent, and it also does not belong to $\mathcal{P}_j$ for any j with $i < j < n$. So p does not belong to J and, hence, (J, J) belongs to $[\![\sim p.]\!]_{\mathsf{RE}} = \mathcal{M}$.

Now suppose that (I, J) belongs to $[\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{RE}}$ and take some $p \in J$. Then,

$$\exists i < n : (p.) \in \mathcal{P}_i \wedge (\forall j : i < j < n \Rightarrow (\sim p.) \notin \mathcal{P}_j) \ ,$$

so, by Corollary 70, $[\![p.]\!]_{\mathsf{RE}}$ belongs to $\langle\!\langle \bigoplus \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}}$. Since (I, J) belongs to $[\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{RE}}$, it must also belong to $[\![p.]\!]_{\mathsf{RE}}$. Thus, p belongs to I and as the choice of $p \in J$ was arbitrary, we can conclude that $I = J$. As a consequence, J is indeed a stable model of $\bigoplus \boldsymbol{P}$.

It remains to prove that J is the only stable model of $\bigoplus \boldsymbol{P}$. Suppose that J' is a stable model of $\bigoplus \boldsymbol{P}$ and take some $p \in J$. We will show that p belongs to J'. We know that

$$\exists i < n : (p.) \in \mathcal{P}_i \wedge (\forall j : i < j < n \Rightarrow (\sim p.) \notin \mathcal{P}_j) \ ,$$

so, by Corollary 70, $[\![p.]\!]_{\mathsf{RE}}$ belongs to $\langle\!\langle \bigoplus \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}}$. Since J' is a stable model of $\bigoplus \boldsymbol{P}$, (J', J') belongs to $[\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{RE}}$ and, consequently, also to $[\![p.]\!]_{\mathsf{RE}}$. Thus, p must belong to J'. Now take some atom p that does not belong to J. We will show that p does not belong to J' either. There are two cases to consider:

a) If $(p.)$ does not belong to $\mathcal{P}_i$ for all $i < n$, then it can be seen that $(J' \setminus \{ p \}, J')$ belongs to all elements of $\langle\!\langle \bigoplus \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}}$. Thus, since J' is a stable model of $\bigoplus \boldsymbol{P}$, $J' \setminus \{ p \} = J'$ and, consequently, p does not belong to J'.

b) If $(p.)$ belongs to $\mathcal{P}_{i_0}$ for some $i_0 < n$ and whenever $(p.)$ belongs to $\mathcal{P}_i$ for some i, there is some j with $i < j < n$ such that $(\sim p.)$ belongs to $\mathcal{P}_j$, then there must exist some i_1 such that $(\sim p.)$ belongs to $\mathcal{P}_{i_1}$ and for all j with $i_1 < j < n$, $(p.)$ does not belong to $\mathcal{P}_j$. Consequently, $[\![\sim p.]\!]_{\mathsf{RE}}$ belongs to $\langle\!\langle \bigoplus \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}}$. Thus, since (J', J') belongs to $[\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{RE}}$, it follows that p cannot belong to J'. $\qquad\square$

Proposition 72. *Let $\boldsymbol{P} = \langle \mathcal{P}, \mathcal{U} \rangle$ be a dynamic logic program, $\oplus$ a δ_{a}-based rule update operator and J an interpretation. If J is a stable model of $\bigoplus \boldsymbol{P}$, then J is a JU-model of $\boldsymbol{P}$.*

Proof. From the assumption we can conclude that (J, J) is an RE-model of $\mathcal{P} \oplus \mathcal{U}$ and for every $I \subsetneq J$, (I, J) is not an RE-model of $\mathcal{P} \oplus \mathcal{U}$.

We need to show that J is a minimal model of the program $\mathcal{Q} = [\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}(\boldsymbol{P}, J)]^J$. First we prove that J is a model of $\mathcal{Q}$. Take some rule $\pi' \in \mathcal{Q}$ and let π be a rule from $[\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}(\boldsymbol{P}, J)]$ such that $\pi' = \pi^J$. We consider two cases:

a) If π belongs to $\mathcal{U}$, then since (J, J) belongs to $[\![\mathcal{P} \oplus \mathcal{U}]\!]_{\mathsf{RE}}$ and $\langle\!\langle \mathcal{P} \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$ contains $[\![\pi]\!]_{\mathsf{RE}}$, (J, J) must also belong to $[\![\pi]\!]_{\mathsf{RE}}$. Thus, J is a model of π and consequently also a model of $\pi' = \pi^J$.

b) If π belongs to $\mathcal{P} \setminus \mathsf{rej}(\boldsymbol{P}, J)$, then since (J, J) belongs to $[\![\mathcal{P} \oplus \mathcal{U}]\!]_{\mathsf{RE}}$ and $\langle\!\langle \mathcal{P} \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$ contains $[\![\pi]\!]_{\mathsf{RE}} \cup \bigcup_{\sigma \in \mathcal{U}} \delta_{\mathsf{a}}([\![\pi]\!]_{\mathsf{RE}}, [\![\sigma]\!]_{\mathsf{RE}})$, (J, J) must also belong to $[\![\pi]\!]_{\mathsf{RE}} \cup \bigcup_{\sigma \in \mathcal{U}} \delta_{\mathsf{a}}([\![\pi]\!]_{\mathsf{RE}}, [\![\sigma]\!]_{\mathsf{RE}})$. Suppose first that for some rule $\sigma \in \mathcal{U}$, (J, J) belongs to the set $\delta_{\mathsf{a}}([\![\pi]\!]_{\mathsf{RE}}, [\![\sigma]\!]_{\mathsf{RE}})$. This implies that $[\![\pi]\!]_{\mathsf{RE}} \bowtie_p^J [\![\sigma]\!]_{\mathsf{RE}}$ for some atom p and by Proposition 64 we can conclude that π belongs to $\mathsf{rej}(\boldsymbol{P}, J)$, contrary to the assumption. Thus, (J, J) does not belong to the set $\bigcup_{\sigma \in \mathcal{U}} \delta_{\mathsf{a}}([\![\pi]\!]_{\mathsf{RE}}, [\![\sigma]\!]_{\mathsf{RE}})$ and, consequently, it belongs to $[\![\pi]\!]_{\mathsf{RE}}$. Hence, J is a model of π and, consequently, it is also a model of $\pi' = \pi^J$.

It remains to prove that J is a minimal model of $\mathcal{Q}$. Take some model I of $\mathcal{Q}$ such that I is a subset of J. We need to show that $I = J$. In the following we will show that (I, J) is a member of the set $[\![\mathcal{P} \oplus \mathcal{U}]\!]_{\mathsf{RE}}$ which, together with the assumption that J is a stable model of $\mathcal{P} \oplus \mathcal{U}$, implies that $I = J$.

So in order to finish the proof, take some set $\mathcal{M}$ from $\langle\!\langle \mathcal{P} \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$. We need to show that (I, J) belongs to $\mathcal{M}$. Recall that $\langle\!\langle \mathcal{P} \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}} = \{ [\![\pi]\!]_{\mathsf{RE}} \cup \bigcup_{\sigma \in \mathcal{U}} \delta_{\mathsf{a}}([\![\pi]\!]_{\mathsf{RE}}, [\![\sigma]\!]_{\mathsf{RE}}) \mid \pi \in \mathcal{P} \} \cup \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$. If $\mathcal{M}$ belongs to $\langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$, then there is some rule $\pi \in \mathcal{U}$ such that $\mathcal{M} = [\![\pi]\!]_{\mathsf{RE}}$. Moreover, π^J belongs to $\mathcal{Q}$, so I is a model of π^J. It then follows that (I, J) is an RE-model of π, i.e. that (I, J) belongs to $\mathcal{M}$, as we wanted to show.

The remaining case is when for some $\pi \in \mathcal{P}$, $\mathcal{M} = [\![\pi]\!]_{\mathsf{RE}} \cup \bigcup_{\sigma \in \mathcal{U}} \delta_{\mathsf{a}}([\![\pi]\!]_{\mathsf{RE}}, [\![\sigma]\!]_{\mathsf{RE}})$. Suppose that (I, J) does not belong to $[\![\pi]\!]_{\mathsf{RE}}$. Then I is not a model of π^J. Since I is a subset of J, we can conclude from this that

$$J \models \mathsf{B}_\pi \ . \tag{B.1}$$

Furthermore, from our assumption that I is a model of $\mathcal{Q}$ it then follows that π^J does not belong to $\mathcal{Q}$ and, consequently, π belongs to $\mathsf{rej}(\boldsymbol{P}, J)$. So there must be some rule $\sigma \in \mathcal{U}$ such that $\mathsf{H}_\sigma = \overline{\mathsf{H}_\pi}$ and $J \models \mathsf{B}_\sigma$. Since we know from the previous part of the proof that J is a model of $\mathcal{Q}$, we can conclude that $J \models \mathsf{H}_\sigma$, so $J \not\models \mathsf{H}_\pi$.

Thus, it follows from (B.1) that J is not a model of π, so (J, J) is not an RE-model of π. But since J is a stable model of $\mathcal{P} \oplus \mathcal{U}$, (J, J) must belong to $\delta_{\mathsf{a}}([\![\pi]\!]_{\mathsf{RE}}, [\![\sigma']\!]_{\mathsf{RE}})$ for some $\sigma' \in \mathcal{U}$ and from the definition of $\delta_{\mathsf{a}}(\cdot, \cdot)$ we obtain that (I, J) also belongs to $\delta_{\mathsf{a}}([\![\pi]\!]_{\mathsf{RE}}, [\![\sigma']\!]_{\mathsf{RE}})$. This implies that (I, J) belongs to $\mathcal{M}$. $\qquad\square$

Proposition 73. *Let $\boldsymbol{P} = \langle \mathcal{P}, \mathcal{U} \rangle$ be a DLP free of local cycles, $\oplus$ a δ_{a}-based rule update operator and J an interpretation. If J is a JU-model of $\boldsymbol{P}$, then J is a stable model of $\bigoplus \boldsymbol{P}$.*

Proof. Suppose that J is a justified update model of $\langle \mathcal{P}, \mathcal{U} \rangle$. Then it is a minimal model of the program $\mathcal{Q} = [\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}(\boldsymbol{P}, J)]^J$. We need to prove that (J, J) is an RE-model of $\mathcal{P} \oplus \mathcal{U}$ and for every $I \subsetneq J$, (I, J) is not an RE-model of $\mathcal{P} \oplus \mathcal{U}$. In order to show that (J, J) is an RE-model of $\mathcal{P} \oplus \mathcal{U}$, recall that

$\langle\!\langle\mathcal{P}\oplus\mathcal{U}\rangle\!\rangle_{\mathsf{RE}} = \big\{\, [\![\pi]\!]_{\mathsf{RE}} \cup \bigcup_{\sigma\in\mathcal{U}}\delta_{\mathsf{a}}([\![\pi]\!]_{\mathsf{RE}},[\![\sigma]\!]_{\mathsf{RE}}) \,\big|\, \pi\in\mathcal{P} \,\big\}\cup\langle\!\langle\mathcal{U}\rangle\!\rangle_{\mathsf{RE}}$ and take some set $\mathcal{M}$ from $\langle\!\langle\mathcal{P}\oplus\mathcal{U}\rangle\!\rangle_{\mathsf{RE}}$. If $\mathcal{M}$ belongs to $\langle\!\langle\mathcal{U}\rangle\!\rangle_{\mathsf{RE}}$, then there is a rule $\pi\in\mathcal{U}$ such that $\mathcal{M}=[\![\pi]\!]_{\mathsf{RE}}$. Also, π^{J} belongs to $\mathcal{Q}$, so J is a model of π^{J}. Consequently, (J,J) belongs to $\mathcal{M}$.

Now suppose that for some π from $\mathcal{P}$, $\mathcal{M}=[\![\pi]\!]_{\mathsf{RE}}\cup\bigcup_{\sigma\in\mathcal{Q}}\delta_{\mathsf{a}}([\![\pi]\!]_{\mathsf{RE}},[\![\sigma]\!]_{\mathsf{RE}})$. If (J,J) does not belong to $[\![\pi]\!]_{\mathsf{RE}}$, then J is not a model of π^{J}, so π belongs to $\mathsf{rej}(\boldsymbol{P},J)$. So there exists a rule σ from $\mathcal{U}$ such that $\mathsf{H}_{\sigma}=\overline{\mathsf{H}_{\pi}}$ and $J\models\mathsf{B}_{\sigma}$. The previous conclusions, together with the fact that π and σ are not local cycles, allow us to use Proposition 64 and conclude that $[\![\pi]\!]_{\mathsf{RE}}\bowtie_{p}^{J}[\![\sigma]\!]_{\mathsf{RE}}$ holds for some atom p. Hence, (J,J) belongs to $\delta_{\mathsf{a}}([\![\pi]\!]_{\mathsf{RE}},[\![\sigma]\!]_{\mathsf{RE}})$, and consequently also to $\mathcal{M}$.

Now suppose that (I,J) belongs to $[\![\pi]\!]_{\mathsf{RE}}$. We will show that I is a model of $\mathcal{Q}$, which implies that $I=J$ because J is by assumption a minimal model of $\mathcal{Q}$. Take some rule π' from $\mathcal{Q}$ and suppose that $\pi'=\pi^{J}$ for some $\pi\in[\mathsf{all}(\boldsymbol{P})\setminus\mathsf{rej}(\boldsymbol{P},J)]$. If π belongs to $\mathcal{U}$, then $[\![\pi]\!]_{\mathsf{RE}}$ belongs to $\langle\!\langle\mathcal{P}\oplus\mathcal{U}\rangle\!\rangle_{\mathsf{RE}}$. Consequently, (I,J) belongs to $[\![\pi]\!]_{\mathsf{RE}}$, so I is a model of π'.

The final case to consider is when π belongs to $\mathcal{P}$. We will prove by contradiction that I is a model of π'. So suppose that I is not a model of π'. Then (I,J) is not an RE-model of π. However, since by assumption (I,J) belongs to $[\![\mathcal{P}\oplus\mathcal{U}]\!]_{\mathsf{RE}}$, it must also belong to the set $[\![\pi]\!]_{\mathsf{RE}}\cup\bigcup_{\sigma\in\mathcal{U}}\delta_{\mathsf{a}}([\![\pi]\!]_{\mathsf{RE}},[\![\sigma]\!]_{\mathsf{RE}})$. We have already shown that it is not a member of $[\![\pi]\!]_{\mathsf{RE}}$, so there must exist some $\sigma\in\mathcal{U}$ such that (I,J) belongs to $\delta_{\mathsf{a}}([\![\pi]\!]_{\mathsf{RE}},[\![\sigma]\!]_{\mathsf{RE}})$. Thus, $[\![\pi]\!]_{\mathsf{RE}}\bowtie_{p}^{J}[\![\sigma]\!]_{\mathsf{RE}}$ holds for some atom p. We can use Proposition 64 to conclude that $\mathsf{H}_{\sigma}=\overline{\mathsf{H}_{\pi}}$ and $J\models\mathsf{B}_{\sigma}$. Hence, π belongs to $\mathsf{rej}(\boldsymbol{P},J)$, contrary to our assumption. $\square$

Proof of Theorem 19. Follows from Propositions 68, 71, 72, 73 and the fact that the JU-semantics has these properties. $\square$

Appendix B.3. Syntactic Properties of δ_{b}- and δ_{c}-Based Operators

Definition 74. We say that $\mathcal{M}\subseteq\mathfrak{X}$ is *RE-rule-expressible* if there exists a rule π such that $\mathcal{M}=[\![\pi]\!]_{\mathsf{RE}}$.

Lemma 75. *Let $\mathcal{M}\subseteq\mathfrak{X}$ be RE-rule-expressible, $\mathcal{S}$ a set of RE-rule-expressible sets of three-valued interpretations, J an interpretation, p an atom, V_0 a truth value and $\delta\in\{\,\delta_{\mathsf{b}},\delta_{\mathsf{c}}\,\}$. If $\mathsf{aug}_{\delta}(\mathcal{M},\mathcal{S})^{J}(p)=\mathsf{V}_0$, then $\mathcal{M}^{J}(p)=\mathsf{V}_0$.*

Proof. Suppose that $\mathsf{aug}_{\delta}(\mathcal{M},\mathcal{S})^{J}(p)=\mathsf{V}_0$. By the definition we then obtain that for all truth values V,

$$J[\mathsf{V}/p]\in\mathsf{aug}_{\delta}(\mathcal{M},\mathcal{S})\text{ if and only if }\mathsf{V}=\mathsf{V}_0\ . \tag{B.2}$$

If the interpretation $J[\mathsf{V}_0/p]$ belongs to $\mathcal{M}$, then we can use (B.2) together with the fact that $\mathcal{M}$ is a subset of $\mathsf{aug}_{\delta}(\mathcal{M},\mathcal{S})$ to conclude that $\mathcal{M}^{J}(p)=\mathsf{V}_0$ and our proof ends.

So suppose that $J[\mathsf{V}_0/p]$ does not belong to $\mathcal{M}$. Then it follows from (B.2) and from the fact that $\mathcal{M}$ is a subset of $\mathsf{aug}_{\delta}(\mathcal{M},\mathcal{S})$ that the interpretations $J[\mathsf{T}/p]$, $J[\mathsf{U}/p]$ and $J[\mathsf{F}/p]$ do not belong to $\mathcal{M}$. Thus, since $J[\mathsf{V}_0/p]$ belongs to $\mathsf{aug}_{\delta}(\mathcal{M},\mathcal{S})$, there must exist some $\mathcal{N}\in\mathcal{S}$ such that $J[\mathsf{V}_0/p]=(I,K)$ belongs to $\delta(\mathcal{M},\mathcal{N})$. In other words, there exists an atom q and an interpretation J' such that $I\subseteq J'\subseteq K$ and $\mathcal{M}^{J'}(q)\neq\mathcal{N}^{J'}(q)$. Note that $J\setminus\{\,p\,\}\subseteq I\subseteq J'\subseteq K\subseteq J\cup\{\,p\,\}$. Thus, $(J',J')=J[\mathsf{V}_1/p]$ for some $\mathsf{V}_1\in\{\,\mathsf{T},\mathsf{F}\,\}$. We distinguish two cases:

   a) If $p\neq q$, then by the definition of δ_{b} and δ_{c} we obtain that $J[\mathsf{V}_1/p],J[\mathsf{U}/p]\in\delta(\mathcal{M},\mathcal{N})$.

   b) If $p=q$, let $\mathsf{V}_2=\mathsf{T}$ is $\mathsf{V}_1=\mathsf{F}$ and $\mathsf{V}_2=\mathsf{F}$ if $\mathsf{V}_1=\mathsf{T}$. It follows that $\mathcal{M}^{J[\mathsf{V}_2/p]}(p)\neq\mathcal{N}^{J[\mathsf{V}_2/p]}(p)$ and by the definition of δ_{b} and δ_{c} we obtain that $J[\mathsf{V}_1/p],J[\mathsf{V}_2/p]\in\delta(\mathcal{M},\mathcal{N})$.

In either case, it is not possible fo $\mathsf{aug}_{\delta}(\mathcal{M},\mathcal{S})^{J}(p)$ to be defined, a conflict with our assumption. $\square$

Proposition 76 (Exception Independence for Rules). *Let $\mathcal{M} \subseteq \mathfrak{X}$ be RE-rule-expressible by a non-disjunctive rule, $\mathcal{S}$, $\mathcal{T}$ be sets of RE-rule-expressible sets of three-valued interpretations and $\delta \in \{\delta_\mathsf{b}, \delta_\mathsf{c}\}$. Then, $\mathsf{aug}_\delta(\mathsf{aug}_\delta(\mathcal{M}, \mathcal{S}), \mathcal{T}) = \mathsf{aug}_\delta(\mathcal{M}, \mathcal{S} \cup \mathcal{T})$.*

Proof. By applying the definition of $\mathsf{aug}_\delta(\cdot, \cdot)$ we can see that our goal is to show that the set

$$\mathsf{aug}_\delta(\mathcal{M}, \mathcal{S}) \cup \bigcup_{\mathcal{N} \in \mathcal{T}} \delta(\mathsf{aug}_\delta(\mathcal{M}, \mathcal{S}), \mathcal{N}) \tag{B.3}$$

is equal to the set $\mathcal{M} \cup \bigcup_{\mathcal{N} \in \mathcal{S} \cup \mathcal{T}} \delta(\mathcal{M}, \mathcal{N})$ which can also be written as

$$\mathsf{aug}_\delta(\mathcal{M}, \mathcal{S}) \cup \bigcup_{\mathcal{N} \in \mathcal{T}} \delta(\mathcal{M}, \mathcal{N}) \ . \tag{B.4}$$

First suppose that some $X \in \mathfrak{X}$ belongs to (B.3). If X belongs to $\mathsf{aug}_\delta(\mathcal{M}, \mathcal{S})$, then it directly follows that X also belongs to (B.4). So suppose that X belongs to $\delta(\mathsf{aug}_\delta(\mathcal{M}, \mathcal{S}), \mathcal{N})$ for some $\mathcal{N} \in \mathcal{T}$. By the definition of δ_b and δ_c we obtain that there exists some atom p and some interpretation J with certain properties relative to X such that $\mathsf{aug}_\delta(\mathcal{M}, \mathcal{S})^J(p) \neq \mathcal{N}^J(p)$. By Lemma 75 we then conclude that $\mathsf{aug}_\delta(\mathcal{M}, \mathcal{S})^J(p) = \mathcal{M}^J(p)$. Thus, X also belongs to $\delta(\mathcal{M}, \mathcal{N})$ and, consequently, also to the set (B.4).

Now suppose that some $X = (I, K) \in \mathfrak{X}$ belongs to (B.4). The case when X belongs to $\mathsf{aug}_\delta(\mathcal{M}, \mathcal{S})$ is trivial, so we assume that X belongs to $\delta(\mathcal{M}, \mathcal{N})$ for some $\mathcal{N} \in \mathcal{T}$. This implies that there exists an atom p and some interpretation J such that $I \subseteq J \subseteq K$ and $\mathcal{M}^J(p) \neq \mathcal{N}^J(p)$. Suppose that $\mathcal{M}^J(p) = \mathsf{V}_0$. If it also holds that $\mathsf{aug}_\delta(\mathcal{M}, \mathcal{S})^J(p) = \mathsf{V}_0$, then it can be seen that X belongs to (B.3). Otherwise it follows from the fact that $\mathcal{M}$ is a subset of $\mathsf{aug}_\delta(\mathcal{M}, \mathcal{S})$ that $\mathsf{aug}_\delta(\mathcal{M}, \mathcal{S})^J(p)$ is undefined and it contains both $J[\mathsf{V}_0/p]$ and $J[\mathsf{V}_1/p] = (I', K')$ for some $\mathsf{V}_1 \neq \mathsf{V}_0$. Thus, for some $\mathcal{N}' \in \mathcal{S}$ it holds that (I', K') belongs to $\delta(\mathcal{M}, \mathcal{N}')$. In other words, there exists an atom q and an interpretation J' such that $I' \subseteq J' \subseteq K'$ and $\mathcal{M}^{J'}(q) \neq \mathcal{N}'^{J'}(q)$. Since $\mathcal{M}$ is expressible by a non-disjunctive rule, it follows from Proposition 63 that $q = p$. Also, J and J' may only differ in the valuation of p, so we obtain that $\mathcal{M}^J(p) \neq \mathcal{N}'^J(p)$. Consequently, X belongs to $\delta(\mathcal{M}, \mathcal{N}')$, so it also belongs to (B.3). $\square$

Definition 77. Let $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ be a DLP. We define $\mathsf{after}_i(\boldsymbol{P}) = \bigcup_{i<j<n} \mathcal{P}_j$.

Proposition 78 (Exception Independence for Programs). *Let $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ be a DLP, $\delta \in \{\delta_\mathsf{b}, \delta_\mathsf{c}\}$ and $\oplus$ a δ-based rule update operator. Then, $\langle\!\langle \bigoplus \boldsymbol{P} \rangle\!\rangle_\mathsf{RE} = \{\mathsf{aug}_\delta(\mathcal{M}, \langle\!\langle \mathsf{after}_i(\boldsymbol{P}) \rangle\!\rangle_\mathsf{RE}) \mid i < n \wedge \mathcal{M} \in \langle\!\langle \mathcal{P}_i \rangle\!\rangle_\mathsf{RE}\}$.*

Proof. By Proposition 66, $\langle\!\langle \bigoplus \boldsymbol{P} \rangle\!\rangle_\mathsf{RE} = \{\mathsf{aug}_\delta(\mathcal{M}, \langle\langle\!\langle \mathcal{P}_j \rangle\!\rangle_\mathsf{RE}\rangle_{i<j<n}) \mid i < n \wedge \mathcal{M} \in \langle\!\langle \mathcal{P}_i \rangle\!\rangle_\mathsf{RE}\}$. The claim follows by induction on n using Proposition 76. $\square$

Proposition 79. *Let $\boldsymbol{P}$ be a DLP and J an interpretation. The following holds:*

(i) *If $\oplus$ is a δ_b-based rule update operator and J is a stable model of $\bigoplus \boldsymbol{P}$, then J is a JU-model of $\boldsymbol{P}$.*

(ii) *If $\oplus$ is a δ_c-based rule update operator and J is a stable model of $\bigoplus \boldsymbol{P}$, then J is a UA-model of $\boldsymbol{P}$.*

Proof. Let $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$. Also, put $\mathsf{rej}(\cdot, \cdot) = \mathsf{rej}_\mathsf{JU}(\cdot, \cdot)$ if $\delta = \delta_\mathsf{b}$ and $\mathsf{rej}(\cdot, \cdot) = \mathsf{rej}_\mathsf{UA}(\cdot, \cdot)$ if $\delta = \delta_\mathsf{c}$. From the assumption we can conclude that (J, J) belongs to $[\![\bigoplus \boldsymbol{P}]\!]_\mathsf{RE}$ and for every $I \subsetneq J$, (I, J) does not belong to $[\![\bigoplus \boldsymbol{P}]\!]_\mathsf{RE}$. We need to show that J is a minimal model of the program $\mathcal{P}' = [\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}(\boldsymbol{P}, J)]^J$.

First we prove that J is a model of $\mathcal{P}'$. Take some rule $\pi' \in \mathcal{P}'$ and let π be a rule from $\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}(\boldsymbol{P}, J)$ such that $\pi' = \pi^J$. Then there is some $i < n$ such that π belongs to $\mathcal{P}_i$. Let $\mathcal{M} = [\![\pi]\!]_\mathsf{RE}$. Since π belongs to $\mathcal{P}_i$, we can use Proposition 78 to conclude that $\langle\!\langle \bigoplus \boldsymbol{P} \rangle\!\rangle_\mathsf{RE}$ contains the set

$$\mathsf{aug}_\delta(\mathcal{M}, \langle\!\langle \mathsf{after}_i(\boldsymbol{P}) \rangle\!\rangle_\mathsf{RE}) = \mathcal{M} \cup \bigcup_{\mathcal{N} \in \langle\!\langle \mathsf{after}_i(\boldsymbol{P}) \rangle\!\rangle_\mathsf{RE}} \delta(\mathcal{M}, \mathcal{N}) \ . \tag{B.5}$$

Furthermore, since (J, J) belongs to $[\![\bigoplus P]\!]_{\mathsf{RE}}$, it must also belong to (B.5). If $(J, J) \in \mathcal{M}$, then $J \models \pi$ and thus also $J \models \pi'$ as desired. So suppose that $(J, J) \notin \mathcal{M}$. Then for some j with $i < j < n$ there exists some $\mathcal{N} \in \langle\!\langle \mathcal{P}_j \rangle\!\rangle_{\mathsf{RE}}$ such that $(J, J) \in \delta(\mathcal{M}, \mathcal{N})$. Thus, for some atom p, $\mathcal{M} \bowtie_p^J \mathcal{N}$, and by Proposition 64 we conclude that there is a rule $\sigma \in \mathcal{P}_j$ with $[\![\sigma]\!]_{\mathsf{RE}} = \mathcal{N}$, $\mathsf{H}_\sigma = \overline{\mathsf{H}_\pi}$ and $J \models \mathsf{B}_\sigma$. We consider two cases:

a) If $\delta = \delta_{\mathsf{b}}$, then the assumption $\pi \notin \mathsf{rej}_{\mathsf{JU}}(P, J)$ is in direct conflict with the existence of σ.

b) If $\delta = \delta_{\mathsf{c}}$, then it follows from $\pi \notin \mathsf{rej}_{\mathsf{UA}}(P, J)$ that $\sigma \in \mathsf{rej}_{\mathsf{UA}}(P, J)$. Thus, there is some rule $\rho \in (\mathcal{P}_k \setminus \mathsf{rej}_{\mathsf{UA}}(P, J))$ with $j < k < n$ such that $\mathsf{H}_\rho = \overline{\mathsf{H}_\sigma} = \mathsf{H}_\pi$ and $J \models \mathsf{B}_\rho$. It follows from $\rho \notin \mathsf{rej}_{\mathsf{UA}}(P, J)$ that no rule from $\mathsf{after}_k(P)$ has the head $\overline{\mathsf{H}_\rho}$ and a body satisfied in J. Thus, the definition of δ_{c} and Proposition 64 imply that the set $\delta_{\mathsf{c}}([\![\rho]\!]_{\mathsf{RE}}, \mathcal{N})$ is empty for all $\mathcal{N} \in \langle\!\langle \mathsf{after}_k(P) \rangle\!\rangle_{\mathsf{RE}}$. But since (J, J) belongs to $[\![\bigoplus P]\!]_{\mathsf{RE}}$ by assumption, it must also belong to the set $[\![\rho]\!]_{\mathsf{RE}} \cup \bigcup_{\mathcal{N} \in \langle\!\langle \mathsf{after}_k(P) \rangle\!\rangle_{\mathsf{RE}}} \delta_{\mathsf{c}}([\![\rho]\!]_{\mathsf{RE}}, \mathcal{N})$ and we conclude that $(J, J) \in [\![\rho]\!]_{\mathsf{RE}}$. Consequently, $J \models \rho$ and from $J \models \mathsf{B}_\rho$ we conclude $J \models \mathsf{H}_\rho$. Since $\mathsf{H}_\pi = \mathsf{H}_\rho$, we have shown that $J \models \pi$ and thus also $J \models \pi'$, as desired.

It remains to prove that J is a minimal model of $\mathcal{P}'$. Take some $I \subseteq J$ such that $I \models \mathcal{P}'$, we need to show that $I = J$. In the following we will show that $(I, J) \in [\![\bigoplus P]\!]_{\mathsf{RE}}$ which, together with the assumption that J is a stable model of $\bigoplus P$, implies that $I = J$. So take some set

$$\mathsf{aug}_\delta(\mathcal{M}, \langle\!\langle \mathsf{after}_i(P) \rangle\!\rangle_{\mathsf{RE}}) = \mathcal{M} \cup \bigcup_{\mathcal{N} \in \langle\!\langle \mathsf{after}_i(P) \rangle\!\rangle_{\mathsf{RE}}} \delta(\mathcal{M}, \mathcal{N}) \ . \tag{B.6}$$

from $\langle\!\langle \bigoplus P \rangle\!\rangle_{\mathsf{RE}}$ with $\mathcal{M} = [\![\pi]\!]_{\mathsf{RE}}$ and $\pi \in \mathcal{P}_i$. We need to show that (I, J) belongs to (B.6). This obviously holds if $(I, J) \in \mathcal{M}$, so suppose that $(I, J) \notin \mathcal{M}$. Then, $I \not\models \pi^J$. Thus, π^J is different from τ and, consequently, $J \models \mathsf{B}_\pi^-$. Also, $\mathsf{B}_\pi^+ \subseteq I$ but $\mathsf{H}_\pi^+ \cap I = \emptyset$ and since $I \subseteq J$, this implies that

$$J \models \mathsf{B}_\pi \ . \tag{B.7}$$

Moreover, since $I \models \mathcal{P}'$, it follows that $\pi^J \notin \mathcal{P}'$, so $\pi \in \mathsf{rej}(P, J)$. Thus, there exists a rule $\sigma \in \mathcal{P}_j$ for some j with $i < j < n$ such that for some atom p and literal $L \in \{ p, \sim p \}$,

$$\mathsf{H}_\pi = \{ L \} \text{ and } \mathsf{H}_\sigma = \{ \overline{L} \} \text{ and } J \models \mathsf{B}_\sigma \ . \tag{B.8}$$

Let $\mathcal{N} = [\![\sigma]\!]_{\mathsf{RE}}$. We consider the following five cases:

a) If $L \in \mathsf{B}_\pi$, then π is tautological and we arrive at a conflict with the assumption $(I, J) \notin [\![\pi]\!]_{\mathsf{RE}}$.

b) If $\overline{L} \in \mathsf{B}_\pi$, then it follows from (B.7) and (B.8) that $(J, J) \notin [\![\pi]\!]_{\mathsf{RE}}$. At the same time, $\delta(\mathcal{M}, \mathcal{N})$ is empty for all $\mathcal{N}$ because for all interpretations J' and atoms q, it is impossible for $\mathcal{M}^{J'}(q)$ to be defined. Thus, we obtain a conflict with the assumption that $(J, J) \in [\![\bigoplus P]\!]_{\mathsf{RE}}$.

c) If $L \in \mathsf{B}_\sigma$, then it follows from (B.8) that $(J, J) \notin [\![\sigma]\!]_{\mathsf{RE}}$. At the same time, $\delta(\mathcal{N}, \mathcal{N}')$ is empty for all $\mathcal{N}'$ because for all interpretations J' and atoms q, it is impossible for $\mathcal{N}^{J'}(q)$ to be defined. Thus, we obtain a conflict with the assumption that $(J, J) \in [\![\bigoplus P]\!]_{\mathsf{RE}}$.

d) If $\overline{L} \in \mathsf{B}_\sigma$, then it follows from (B.8) that $J \models \overline{L}$ and together with (B.7) we obtain that $J \not\models \pi$. Thus, $(J, J) \notin [\![\pi]\!]_{\mathsf{RE}}$ and since $(J, J) \in [\![\bigoplus P]\!]_{\mathsf{RE}}$, there must exist some $\mathcal{N}' \in \langle\!\langle \mathsf{after}_i(P) \rangle\!\rangle_{\mathsf{RE}}$ such that $(J, J) \in \delta(\mathcal{M}, \mathcal{N}')$. It follows from the definitions of δ_{b} and δ_{c} that $(I, J) \in \delta(\mathcal{M}, \mathcal{N}')$, and thus also $(I, J) \in \mathsf{aug}_\delta(\mathcal{M}, \langle\!\langle \mathsf{after}_i(P) \rangle\!\rangle_{\mathsf{RE}})$.

e) In the remaining case, B_π and B_σ do not contain L nor $\overline{L}$. Thus, we can use Proposition 64 to conclude that $\mathcal{M} \bowtie_p^J \mathcal{N}$. Note that if $\delta = \delta_{\mathsf{c}}$, then the definition of $\mathsf{rej}_{\mathsf{UA}}(\cdot, \cdot)$ implies that $\sigma \notin$

$\mathsf{rej}_{\mathsf{UA}}(\boldsymbol{P}, J)$, so $J \models \sigma$. Using (B.8) and (B.7) we can conclude that $J \not\models \pi$, so $(J, J) \notin \mathcal{M}$. Consequently, by the definitions of δ_{b} and δ_{c} it follows that $(I, J) \in \delta(\mathcal{M}, \mathcal{N})$, and thus also $(I, J) \in \mathsf{aug}_{\delta}(\mathcal{M}, \langle\!\langle \mathsf{after}_i(\boldsymbol{P}) \rangle\!\rangle_{\mathsf{RE}})$. $\qquad\square$

Proposition 80. *Let $\boldsymbol{P}$ be a DLP free of local cycles and J an interpretation. The following holds:*

(i) If $\oplus$ is a δ_{b}-based rule update operator and J is a JU-model of $\boldsymbol{P}$, then J is a stable model of $\bigoplus \boldsymbol{P}$.

(ii) If $\oplus$ is a δ_{c}-based rule update operator and J is a UA-model of $\boldsymbol{P}$, then J is a stable model of $\bigoplus \boldsymbol{P}$.

Proof. Let $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$. Also, put $\mathsf{rej}(\cdot, \cdot) = \mathsf{rej}_{\mathsf{JU}}(\cdot, \cdot)$ if $\delta = \delta_{\mathsf{b}}$ and $\mathsf{rej}(\cdot, \cdot) = \mathsf{rej}_{\mathsf{UA}}(\cdot, \cdot)$ if $\delta = \delta_{\mathsf{c}}$. From the assumption we can conclude that J is a minimal model of the program $\mathcal{P}' = [\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}(\boldsymbol{P}, J)]^J$. We need to prove that $(J, J) \in [\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{RE}}$ and for every $I \subsetneq J$, $(I, J) \notin [\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{RE}}$.

In order to show that $(J, J) \in [\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{RE}}$, take some set

$$\mathsf{aug}_{\delta}(\mathcal{M}, \langle\!\langle \mathsf{after}_i(\boldsymbol{P}) \rangle\!\rangle_{\mathsf{RE}}) = \mathcal{M} \cup \bigcup_{\mathcal{N} \in \langle\!\langle \mathsf{after}_i(\boldsymbol{P}) \rangle\!\rangle_{\mathsf{RE}}} \delta(\mathcal{M}, \mathcal{N}) \tag{B.9}$$

from $\langle\!\langle \bigoplus \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}}$ where $\mathcal{M} = [\![\pi]\!]_{\mathsf{RE}}$ for some $\pi \in \mathcal{P}_i$. If (J, J) belongs to $\mathcal{M}$, then it obviously belongs to (B.9). So assume that $(J, J) \notin \mathcal{M}$. Then $J \not\models \pi$, so

$$J \models \mathsf{B}_{\pi} \ , \tag{B.10}$$

and we can also conclude that π belongs to $\mathsf{rej}(\boldsymbol{P}, J)$. As a consequence, there exists a rule σ from $\mathcal{P}_j$ for some j with $i < j < n$ such that for some atom p and literal $L \in \{ p, \sim p \}$,

$$\mathsf{H}_{\pi} = \{ L \} \text{ and } \mathsf{H}_{\sigma} = \{ \overline{L} \} \text{ and } J \models \mathsf{B}_{\sigma} \ . \tag{B.11}$$

Let $\mathcal{N} = [\![\sigma]\!]_{\mathsf{RE}}$. It can be verified that (B.10) and (B.11), together with the assumption that π and σ are not local cycles, allow us to use Proposition 64 and conclude that for some atom p, $\mathcal{M} \bowtie_p^J \mathcal{N}$. Thus, (J, J) belongs to $\delta(\mathcal{M}, \mathcal{N})$, and consequently also to (B.9).

Now suppose that $(I, J) \in [\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{RE}}$. We will show that $I \models \mathcal{P}'$, which implies that $I = J$ because J is by assumption a minimal model of $\mathcal{P}'$. Take some $\pi' \in \mathcal{P}'$. Then there is a rule π from $\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}(\boldsymbol{P}, J)$ such that $\pi' = \pi^J$. Suppose that $\pi \in \mathcal{P}_i$. We will prove by contradiction that $I \models \pi'$. So suppose that $I \not\models \pi'$. Then π^J is different from τ and, consequently, $J \models \mathsf{B}_{\pi}^-$. Also, B_{π}^+ is included in I, so since I is a subset of J, B_{π}^+ is included in J as well. Hence,

$$J \models \mathsf{B}_{\pi} \ . \tag{B.12}$$

Also, $(I, J) \notin [\![\pi]\!]_{\mathsf{RE}}$. By our assumption, $(I, J) \in \mathcal{M} \cup \bigcup_{\mathcal{N} \in \langle\!\langle \mathsf{after}_i(\boldsymbol{P}) \rangle\!\rangle_{\mathsf{RE}}} \delta(\mathcal{M}, \mathcal{N})$ where $\mathcal{M} = [\![\pi]\!]_{\mathsf{RE}}$. We have already shown that $(I, J) \notin \mathcal{M}$, so there must be a rule $\sigma \in \mathcal{P}_j$ for some j with $i < j < n$ such that $(I, J) \in \delta(\mathcal{M}, \mathcal{N})$ where $\mathcal{N} = [\![\sigma]\!]_{\mathsf{RE}}$. Thus, there exists some interpretation K and an atom p such that $I \subseteq K \subseteq J$, $\mathcal{M} \bowtie_p^K \mathcal{N}$ and if $p \in J \setminus I$, then $J = K$. By Proposition 64, there is a literal $L \in \{ p, \sim p \}$ such that $\mathsf{H}_{\pi} = \{ L \}$ and $\mathsf{H}_{\sigma} = \{ \overline{L} \}$. We consider three cases:

a) If $p \notin J \setminus I$, then $I(p) = J(p)$. Furthermore, $J \models \pi$ because $J \models \mathcal{P}'$ and from (B.12) we obtain that $J \models \mathsf{H}_{\pi}$. Thus, $I \models \pi^J$, a conflict with the assumption that $I \not\models \pi'$.

b) If $p \in J \setminus I$ and $\delta = \delta_{\mathsf{b}}$, then it follows from the definition of δ_{b} that $J = K$, so $\mathcal{M} \bowtie_p^J \mathcal{N}$. Thus, by Proposition 64 we conclude that $\pi \in \mathsf{rej}(\boldsymbol{P}, J)$, contrary to the way π was picked.

c) If $p \in J \setminus I$ and $\delta = \delta_{\mathsf{c}}$, then the definition of δ_{c} implies that $(J, J) \notin [\![\pi]\!]_{\mathsf{RE}}$, contrary to the assumption that $J \models \mathcal{P}'$. $\qquad\square$

46

Appendix B.4. Syntactic Properties of δ_{d}- and δ_{e}-Based Operators

Lemma 81. *Let P be a DLP, $\oplus_{\mathsf{b}}$, $\oplus_{\mathsf{c}}$, $\oplus_{\mathsf{d}}$ and $\oplus_{\mathsf{e}}$ be δ_{b}-, δ_{c}-, δ_{d}- and δ_{e}-based rule update operators, respectively. Then the following holds:*

   (i) If $\mathcal{M} \in \langle\!\langle \bigoplus_{\mathsf{d}} P \rangle\!\rangle_{\mathsf{RE}}$, then either $\mathcal{M} = \mathcal{X}$ or $\mathcal{M} \in \langle\!\langle \bigoplus_{\mathsf{b}} P \rangle\!\rangle_{\mathsf{RE}}$;

   (ii) If $\mathcal{M} \in \langle\!\langle \bigoplus_{\mathsf{e}} P \rangle\!\rangle_{\mathsf{RE}}$, then either $\mathcal{M} = \mathcal{X}$ or $\mathcal{M} \in \langle\!\langle \bigoplus_{\mathsf{c}} P \rangle\!\rangle_{\mathsf{RE}}$.

Proof. Follows by induction on the length of P. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\square$

Lemma 82. *Let P be a DLP, $\oplus_{\mathsf{b}}$, $\oplus_{\mathsf{c}}$, $\oplus_{\mathsf{d}}$ and $\oplus_{\mathsf{e}}$ be δ_{b}-, δ_{c}-, δ_{d}- and δ_{e}-based rule update operators, respectively. Then the following holds:*

   (i) If $\mathcal{M} \in \langle\!\langle \bigoplus_{\mathsf{b}} P \rangle\!\rangle_{\mathsf{RE}}$, then for some $\mathcal{N} \in \langle\!\langle \bigoplus_{\mathsf{d}} P \rangle\!\rangle_{\mathsf{RE}}$, $\mathcal{N} \subseteq \mathcal{M}$;

   (ii) If $\mathcal{M} \in \langle\!\langle \bigoplus_{\mathsf{c}} P \rangle\!\rangle_{\mathsf{RE}}$, then for some $\mathcal{N} \in \langle\!\langle \bigoplus_{\mathsf{e}} P \rangle\!\rangle_{\mathsf{RE}}$, $\mathcal{N} \subseteq \mathcal{M}$.

Proof. The following two stronger statements follow by induction on the length of P using Proposition 76:

   (i) If $\mathcal{M} \in \langle\!\langle \bigoplus_{\mathsf{b}} P \rangle\!\rangle_{\mathsf{RE}}$, then for some set of RE-rule-expressible sets of three-valued interpretations $\mathcal{S}$ and some $\mathcal{N} \in \langle\!\langle \bigoplus_{\mathsf{d}} P \rangle\!\rangle_{\mathsf{RE}}$, $\mathcal{M} = \mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{N}, \mathcal{S})$.

   (ii) If $\mathcal{M} \in \langle\!\langle \bigoplus_{\mathsf{c}} P \rangle\!\rangle_{\mathsf{RE}}$, then for some set of RE-rule-expressible sets of three-valued interpretations $\mathcal{S}$ and some $\mathcal{N} \in \langle\!\langle \bigoplus_{\mathsf{e}} P \rangle\!\rangle_{\mathsf{RE}}$, $\mathcal{M} = \mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{N}, \mathcal{S})$. $\qquad\qquad\square$

Proposition 83. *Let P be a DLP, $\oplus_{\mathsf{b}}$, $\oplus_{\mathsf{c}}$, $\oplus_{\mathsf{d}}$ and $\oplus_{\mathsf{e}}$ be δ_{b}-, δ_{c}-, δ_{d}- and δ_{e}-based rule update operators, respectively. Then $[\![\bigoplus_{\mathsf{b}} P]\!]_{\mathsf{SM}} = [\![\bigoplus_{\mathsf{d}} P]\!]_{\mathsf{SM}}$ and $[\![\bigoplus_{\mathsf{c}} P]\!]_{\mathsf{SM}} = [\![\bigoplus_{\mathsf{e}} P]\!]_{\mathsf{SM}}$.*

Proof. By Lemma 81, $\langle\!\langle \bigoplus_{\mathsf{b}} P \rangle\!\rangle_{\mathsf{RE}} \cup \{\, \mathcal{X} \,\} \supseteq \langle\!\langle \bigoplus_{\mathsf{d}} P \rangle\!\rangle_{\mathsf{RE}}$ and $\langle\!\langle \bigoplus_{\mathsf{c}} P \rangle\!\rangle_{\mathsf{RE}} \cup \{\, \mathcal{X} \,\} \supseteq \langle\!\langle \bigoplus_{\mathsf{e}} P \rangle\!\rangle_{\mathsf{RE}}$, so

$$[\![\textstyle\bigoplus_{\mathsf{b}} P]\!]_{\mathsf{RE}} = \bigcap \langle\!\langle \textstyle\bigoplus_{\mathsf{b}} P \rangle\!\rangle_{\mathsf{RE}} = \bigcap (\langle\!\langle \textstyle\bigoplus_{\mathsf{b}} P \rangle\!\rangle_{\mathsf{RE}} \cup \{\, \mathcal{X} \,\}) \subseteq \bigcap \langle\!\langle \textstyle\bigoplus_{\mathsf{d}} P \rangle\!\rangle_{\mathsf{RE}} = [\![\textstyle\bigoplus_{\mathsf{d}} P]\!]_{\mathsf{RE}} \ ,$$
$$[\![\textstyle\bigoplus_{\mathsf{c}} P]\!]_{\mathsf{RE}} = \bigcap \langle\!\langle \textstyle\bigoplus_{\mathsf{c}} P \rangle\!\rangle_{\mathsf{RE}} = \bigcap (\langle\!\langle \textstyle\bigoplus_{\mathsf{c}} P \rangle\!\rangle_{\mathsf{RE}} \cup \{\, \mathcal{X} \,\}) \subseteq \bigcap \langle\!\langle \textstyle\bigoplus_{\mathsf{e}} P \rangle\!\rangle_{\mathsf{RE}} = [\![\textstyle\bigoplus_{\mathsf{e}} P]\!]_{\mathsf{RE}} \ .$$

Furthermore, by Lemma 82,

$$[\![\textstyle\bigoplus_{\mathsf{b}} P]\!]_{\mathsf{RE}} = \bigcap \langle\!\langle \textstyle\bigoplus_{\mathsf{b}} P \rangle\!\rangle_{\mathsf{RE}} = \bigcap_{\mathcal{M} \in \langle\!\langle \oplus_{\mathsf{b}} P \rangle\!\rangle_{\mathsf{RE}}} \mathcal{M} \supseteq \bigcap_{\mathcal{N} \in \langle\!\langle \oplus_{\mathsf{d}} P \rangle\!\rangle_{\mathsf{RE}}} \mathcal{N} = \bigcap \langle\!\langle \textstyle\bigoplus_{\mathsf{d}} P \rangle\!\rangle_{\mathsf{RE}} = [\![\textstyle\bigoplus_{\mathsf{d}} P]\!]_{\mathsf{RE}} \ ,$$
$$[\![\textstyle\bigoplus_{\mathsf{c}} P]\!]_{\mathsf{RE}} = \bigcap \langle\!\langle \textstyle\bigoplus_{\mathsf{c}} P \rangle\!\rangle_{\mathsf{RE}} = \bigcap_{\mathcal{M} \in \langle\!\langle \oplus_{\mathsf{c}} P \rangle\!\rangle_{\mathsf{RE}}} \mathcal{M} \supseteq \bigcap_{\mathcal{N} \in \langle\!\langle \oplus_{\mathsf{e}} P \rangle\!\rangle_{\mathsf{RE}}} \mathcal{N} = \bigcap \langle\!\langle \textstyle\bigoplus_{\mathsf{e}} P \rangle\!\rangle_{\mathsf{RE}} = [\![\textstyle\bigoplus_{\mathsf{e}} P]\!]_{\mathsf{RE}} \ .$$

Thus, $[\![\bigoplus_{\mathsf{b}} P]\!]_{\mathsf{RE}} = [\![\bigoplus_{\mathsf{d}} P]\!]_{\mathsf{RE}}$ and $[\![\bigoplus_{\mathsf{c}} P]\!]_{\mathsf{RE}} = [\![\bigoplus_{\mathsf{e}} P]\!]_{\mathsf{RE}}$, and the rest follows from Proposition 15. $\quad\square$

Proof of Theorem 23. Follows from Propositions 79, 80 and 83. $\qquad\qquad\qquad\qquad\qquad\square$

Proof of Theorem 22. Follows from Theorem 23 and from the fact that the JU- and UA-models satisfy all of these properties. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\square$

Proof of Corollary 24. It suffices to put $\mathcal{R} = \bigoplus \langle \mathcal{P}_i \rangle_{i \leq j}$ and apply Theorem 23. $\qquad\qquad\qquad\square$

Appendix B.5. Semantic Properties

Proposition 84. *Let $\oplus$ be a δ-based rule update operator. Then $\oplus$ satisfies (Initialisation), (Disjointness), (P1) and (P2.$\top$) with respect to RR, SR, RMR, SMR, RE, SE and SM (where applicable).*

Proof. We prove these properties with respect to RR-equivalence; their satisfaction with respect to the other notions of program equivalence follows from Proposition 15. To verify that (Initialisation) holds, it suffices to observe that

$$\langle\!\langle \emptyset \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}} = \{\, \mathsf{aug}_\delta(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}) \mid \mathcal{M} \in \emptyset \,\} \cup \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}} = \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}} \ .$$

Thus, $\emptyset \oplus \mathcal{U}$ is RR-equivalent to $\mathcal{U}$. As for (Disjointness), it suffices to observe that $\langle\!\langle (\mathcal{R} \cup \mathcal{S}) \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$ coincides with

$$\{\, \mathsf{aug}_\delta\,(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}) \mid \mathcal{M} \in \mathcal{R} \cup \mathcal{S} \,\} \cup \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$$
$$= (\{\, \mathsf{aug}_\delta\,(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}) \mid \mathcal{M} \in \mathcal{R} \,\} \cup \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}) \cup (\{\, \mathsf{aug}_\delta\,(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}) \mid \mathcal{M} \in \mathcal{S} \,\} \cup \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}})$$
$$= \langle\!\langle \mathcal{R} \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}} \cup \langle\!\langle \mathcal{S} \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}} = \langle\!\langle (\mathcal{R} \oplus \mathcal{U}) \cup (\mathcal{S} \oplus \mathcal{U}) \rangle\!\rangle_{\mathsf{RE}} \ .$$

In order to prove that (P1) holds, consider that $\langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$ is a subset of $\langle\!\langle \mathcal{R} \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$. Consequently, $\mathcal{R} \oplus \mathcal{U} \models_{\mathsf{RR}} \mathcal{U}$. Finally, (P2.$\top$) follows from the fact that $\mathsf{aug}_\delta(\mathcal{M}, \emptyset) = \mathcal{M}$ for all $\mathcal{M} \subseteq \mathcal{X}$. □

Lemma 85. *Let $\mathcal{P}$, $\mathcal{Q}$ be programs over disjoint alphabets and $\delta \in \{\, \delta_\mathsf{a}, \delta_\mathsf{b}, \delta_\mathsf{c}, \delta_\mathsf{d}, \delta_\mathsf{e} \,\}$. Then for all $\mathcal{M} \in \langle\!\langle \mathcal{P} \rangle\!\rangle_{\mathsf{RE}}$, either $\mathsf{aug}_\delta(\mathcal{M}, \langle\!\langle \mathcal{Q} \rangle\!\rangle_{\mathsf{RE}}) = \mathcal{M}$ or $\mathsf{aug}_\delta(\mathcal{M}, \langle\!\langle \mathcal{Q} \rangle\!\rangle_{\mathsf{RE}}) = \mathcal{X}$.*

Proof. Follows from Proposition 63 and the definitions of δ_a, δ_b, δ_c, δ_d, and δ_e. □

Lemma 86. *Let $\mathcal{M} \subseteq \mathcal{X}$ be RE-rule-expressible, $\mathcal{U}$ a program, p an atom and V_0 a truth value. If $\mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}})^J(p) = \mathsf{V}_0$, then either $\mathcal{M}^J(p) = \mathsf{V}_0$, or p occurs in $\mathcal{U}$.*

Proof. Suppose that $\mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}})^J(p) = \mathsf{V}_0$ and $\mathcal{M}^J(p) \neq \mathsf{V}_0$. Thus, $\mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}})$ contains $J[\mathsf{V}/p]$ if and only if $\mathsf{V} = \mathsf{V}_0$. Since $\mathcal{M}$ is a subset of $\mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}})$, it follows that $\mathcal{M}$ does not contain $J[\mathsf{T}/p]$, $J[\mathsf{U}/p]$ nor $J[\mathsf{F}/p]$. Let π be some rule such that $[\![\pi]\!]_{\mathsf{RE}} = \mathcal{M}$. We can conclude that p does not occur in π. Furthermore, if $\mathsf{V}_0 = \mathsf{T}$, then by the definition of δ_a we obtain a conflict with the fact that $J[\mathsf{U}/p]$ does not belong to $\mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}})$. Consequently, $\mathsf{V}_0 = \mathsf{F}$. Furthermore, since $J[\mathsf{T}/p]$ does not belong to $\mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}})$ but $J[\mathsf{F}/p]$ does, there exists some $\mathcal{N} \in \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$ and some atom q such that $\mathcal{M} \bowtie_q^{J[\mathsf{F}/p]} \mathcal{N}$ but it is not the case that $\mathcal{M} \bowtie_q^{J[\mathsf{T}/p]} \mathcal{N}$. Since p does not occur in π, this is only possible if p occurs in $\mathcal{U}$. □

Proposition 87. *Let $\oplus$ be a δ_a-, δ_b-, δ_c, δ_d or δ_e-based rule update operator. Then $\oplus$ satisfies (Non-interference) for non-disjunctive programs with respect to RR, SR, RMR, SMR, RE, SE and SM.*

Proof. We prove this property with respect to RR-equivalence; its satisfaction with respect to the other notions of program equivalence follows from Proposition 15. Suppose that $\mathcal{P}, \mathcal{U}$ and $\mathcal{V}$ are non-disjunctive programs, $\delta \in \{\, \delta_\mathsf{a}, \delta_\mathsf{b}, \delta_\mathsf{c}, \delta_\mathsf{d}, \delta_\mathsf{e} \,\}$ and $\oplus$ is a δ-based rule update operator. Take some $\mathcal{M}_0 \in \langle\!\langle (\mathcal{P} \oplus \mathcal{U}) \oplus \mathcal{V} \rangle\!\rangle_{\mathsf{RE}}$. We will show that $\mathcal{M}_0$ either belongs to $\langle\!\langle (\mathcal{P} \oplus \mathcal{V}) \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$ or $\mathcal{M}_0 = \mathcal{X}$. Note that $\langle\!\langle (\mathcal{P} \oplus \mathcal{U}) \oplus \mathcal{V} \rangle\!\rangle_{\mathsf{RE}}$ coincides with $\{\, \mathsf{aug}_\delta\,(\mathcal{M}, \langle\!\langle \mathcal{V} \rangle\!\rangle_{\mathsf{RE}}) \mid \mathcal{M} \in \langle\!\langle \mathcal{P} \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}} \,\} \cup \langle\!\langle \mathcal{V} \rangle\!\rangle_{\mathsf{RE}}$. We consider three cases:

a) Suppose that $\mathcal{M}_0 \in \langle\!\langle \mathcal{V} \rangle\!\rangle_{\mathsf{RE}}$. Then $\mathcal{M}_0 \in \langle\!\langle \mathcal{P} \oplus \mathcal{V} \rangle\!\rangle_{\mathsf{RE}}$ and since $\langle\!\langle (\mathcal{P} \oplus \mathcal{V}) \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$ coincides with $\{\, \mathsf{aug}_\delta(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}) \mid \mathcal{M} \in \langle\!\langle \mathcal{P} \oplus \mathcal{V} \rangle\!\rangle_{\mathsf{RE}} \,\} \cup \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$, it must contain the set $\mathsf{aug}_\delta(\mathcal{M}_0, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}})$. Furthermore, since $\mathcal{U}$ and $\mathcal{V}$ are over disjoint alphabets, it follows from Lemma 85 that $\mathsf{aug}_\delta(\mathcal{M}_0, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}})$ is either $\mathcal{M}_0$ or $\mathcal{X}$, as desired.

b) Suppose that $\mathcal{M}_0 = \mathsf{aug}_\delta(\mathcal{M}, \langle\!\langle \mathcal{V} \rangle\!\rangle_{\mathsf{RE}})$ and $\mathcal{M} \in \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$. As in the previous case, since $\mathcal{U}$ and $\mathcal{V}$ are over disjoint alphabets, it follows from Lemma 85 that $\mathsf{aug}_\delta(\mathcal{M}, \langle\!\langle \mathcal{V} \rangle\!\rangle_{\mathsf{RE}})$ is either $\mathcal{M}$ or $\mathfrak{X}$. If the former is true, then $\mathcal{M}_0 \in \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$, so $\mathcal{M}_0 \in \langle\!\langle (\mathcal{P} \oplus \mathcal{V}) \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$.

c) Suppose that $\mathcal{M}_0 = \mathsf{aug}_\delta\left(\mathsf{aug}_\delta\left(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}\right), \langle\!\langle \mathcal{V} \rangle\!\rangle_{\mathsf{RE}}\right)$ for some $\mathcal{M} \in \langle\!\langle \mathcal{P} \rangle\!\rangle_{\mathsf{RE}}$. If $\delta \in \{\, \delta_\mathsf{b}, \delta_\mathsf{c} \,\}$, then it follows directly from Proposition 76 that

$$\mathcal{M}_0 = \mathsf{aug}_\delta\left(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}} \cup \langle\!\langle \mathcal{V} \rangle\!\rangle_{\mathsf{RE}}\right) = \mathsf{aug}_\delta\left(\mathsf{aug}_\delta\left(\mathcal{M}, \langle\!\langle \mathcal{V} \rangle\!\rangle_{\mathsf{RE}}\right), \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}\right) \ .$$

Consequently, $\mathcal{M}_0 \in \langle\!\langle (\mathcal{P} \oplus \mathcal{V}) \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$.
If $\delta \in \{\, \delta_\mathsf{d}, \delta_\mathsf{e} \,\}$, then either $\mathcal{M}_0 = \mathfrak{X}$ or $\mathcal{M}_0 = \mathsf{aug}_{\delta_\mathsf{b}}\left(\mathsf{aug}_{\delta_\mathsf{b}}\left(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}\right), \langle\!\langle \mathcal{V} \rangle\!\rangle_{\mathsf{RE}}\right)$. The rest follows by the previous paragraph.
Finally, if $\delta = \delta_\mathsf{a}$, then we consider two subcases:

   (i) If $\mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}) = \mathcal{M}$, then $\mathcal{M}_0 = \mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}, \langle\!\langle \mathcal{V} \rangle\!\rangle_{\mathsf{RE}})$, so $\mathcal{M}_0$ belongs to $\langle\!\langle \mathcal{P} \oplus \mathcal{V} \rangle\!\rangle_{\mathsf{RE}}$. Take some $\mathcal{N} \in \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$ and suppose that $\mathcal{M}_0 \bowtie^p_J \mathcal{N}$. Then, by Lemma 86, either $\mathcal{M}^J(p) = \mathsf{V}_0$ for some truth value V_0, or p occurs in $\mathcal{V}$. In the former case we obtain a conflict with the assumption that $\mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}) = \mathcal{M}$ while the latter case is in conflict with the assumption that $\mathcal{U}$ and $\mathcal{V}$ are over disjoint alphabets. Thus, no such $\mathcal{N}$ exists and $\mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}_0, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}) = \mathcal{M}_0$. Consequently, $\mathcal{M}_0$ belongs to $\langle\!\langle (\mathcal{P} \oplus \mathcal{V}) \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$.

   (ii) If $\mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}) \neq \mathcal{M}$, then there is some $\mathcal{N} \in \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$ such that $\mathcal{M} \bowtie^p_J \mathcal{N}$ for some atom p and interpretation J. Thus since $\mathcal{U}$ and $\mathcal{V}$ are over disjoint alphabets, it follows from Proposition 64 that $\mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}, \langle\!\langle \mathcal{V} \rangle\!\rangle_{\mathsf{RE}}) = \mathcal{M}$, so $\mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}})$ belongs to $\langle\!\langle (\mathcal{P} \oplus \mathcal{V}) \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$. It remains to show that $\mathcal{M}_0 = \mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}})$. Put $\mathcal{M}_1 = \mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}})$ and suppose that for some $\mathcal{N}' \in \langle\!\langle \mathcal{V} \rangle\!\rangle_{\mathsf{RE}}$, some atom q and some interpretation K, $\mathcal{M}_1 \bowtie^K_q \mathcal{N}'$. Then, by Lemma 86, either $\mathcal{M}^K(q) = \mathsf{V}_0$ for some truth value V_0, or q occurs in $\mathcal{U}$. In the former case, $p = q$ by Proposition 64 and we obtain a conflict with the fact that $\mathcal{U}$ and $\mathcal{V}$ are over disjoint alphabets. In the latter case, q occurs in both $\mathcal{U}$ and $\mathcal{V}$, so the same conflict follows. Consequently, no such $\mathcal{N}'$ exists and $\mathcal{M}_0 = \mathsf{aug}_{\delta_\mathsf{a}}\left(\mathsf{aug}_{\delta_\mathsf{a}}\left(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}\right), \langle\!\langle \mathcal{V} \rangle\!\rangle_{\mathsf{RE}}\right) = \mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}_1, \langle\!\langle \mathcal{V} \rangle\!\rangle_{\mathsf{RE}}) = \mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}})$.

The proof of the other inclusion is symmetric. $\qquad\square$

Remark 88 ((Non-interference) for Disjunctive Programs)**.** The following programs form a counterexample to (Non-interference) for δ_b-, δ_c-, δ_d- and δ_e-based operators under SM-equivalence (and thus under all stronger notions of equivalence as well):

$$\mathcal{P}: \quad p; q; r. \qquad\qquad \mathcal{U}: \quad \sim p \leftarrow \sim r. \qquad\qquad \mathcal{V}: \quad \sim q.$$
$$p \leftarrow r.$$
$$r \leftarrow p.$$

As for δ_a-based operators, the singleton programs $\mathcal{P}' = \{\, p; q. \,\}$, $\mathcal{U}' = \{\, \sim p \leftarrow \sim r. \,\}$ and $\mathcal{V}' = \{\, \sim q \leftarrow s. \,\}$ form a counterexample to (Non-interference) under SE-equivalence and all stronger notions of equivalence.

Proposition 89. *Let $\oplus$ be a δ-based rule update operator where $\delta(\mathcal{M}, \mathfrak{X}) \subseteq \mathcal{M}$ for all $\mathcal{M} \subseteq \mathfrak{X}$. Then $\oplus$ satisfies (Tautology) and (Immunity to Tautologies) with respect to RR, SR, RMR, SMR, RE, SE and SM.*

Proof. For RR-equivalence this can be verified in a straight-forward manner. For the remaining notions of program equivalence this follows from Proposition 15. $\qquad\square$

Proposition 90. *Let $\oplus$ be a δ-based rule update operator. Then $\oplus$ satisfies* (Idempotence) *with respect to RMR, SMR, RE, SE and SM. Moreover, if $\oplus$ is δ_{d}- or δ_{e}-based, then it also satisfies* (Idempotence) *with respect to RR and SR.*

Proof. (Idempotence) states the following: $\mathcal{R} \oplus \mathcal{R} \equiv \mathcal{R}$. We will show that this is true under RMR-equivalence which, together with Proposition 15, implies that it holds under SMR-, RE-, SE- and SM-equivalence.

First take some $\mathcal{M} \in \min\langle\!\langle (\mathcal{R} \oplus \mathcal{R}) \cup \{\,\tau\,\} \rangle\!\rangle_{\mathsf{RE}}$. If $\mathcal{M} = \mathcal{X}$, then $\mathcal{R} \oplus \mathcal{R}$ is tautological and since $\langle\!\langle \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$ is a subset of $\langle\!\langle \mathcal{R} \oplus \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$, $\mathcal{R}$ is itself tautological. Thus, $\mathcal{X}$ also belongs to $\min\langle\!\langle \mathcal{R} \cup \{\,\tau\,\} \rangle\!\rangle_{\mathsf{RE}}$. In the principal case, either $\mathcal{M} \in \langle\!\langle \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$, or $\mathcal{M} = \mathsf{aug}_\delta(\mathcal{M}_0, \langle\!\langle \mathcal{R} \rangle\!\rangle_{\mathsf{RE}})$ for some $\mathcal{M}_0 \in \langle\!\langle \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$. In the latter case we have that $\mathcal{M}_0$ is a subset of $\mathcal{M}$ and since $\mathcal{M}_0$ belongs to $\langle\!\langle \mathcal{R} \oplus \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$, by the minimality of $\mathcal{M}$ we obtain that $\mathcal{M} = \mathcal{M}_0$, so $\mathcal{M}$ belongs to $\langle\!\langle \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$. Now it follows that $\mathcal{M}$ is minimal $\langle\!\langle \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$ because $\langle\!\langle \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$ is a subset of $\langle\!\langle \mathcal{R} \oplus \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$ and $\mathcal{M}$ is minimal in the latter set.

Now take some $\mathcal{M} \in \min\langle\!\langle \mathcal{R} \cup \{\,\tau\,\} \rangle\!\rangle_{\mathsf{RE}}$. If $\mathcal{M} = \mathcal{X}$, then $\mathcal{R}$ is tautological and it follows by the properties of $\oplus$ that $\mathcal{R} \oplus \mathcal{R}$ is also tautological. Thus, $\mathcal{X}$ also belongs to $\langle\!\langle (\mathcal{R} \oplus \mathcal{R}) \cup \{\,\tau\,\} \rangle\!\rangle_{\mathsf{RE}}$. In the principal case, $\mathcal{M} \in \langle\!\langle \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$. Take some $\mathcal{N} \in \langle\!\langle \mathcal{R} \oplus \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$ such that $\mathcal{N}$ is a subset of $\mathcal{M}$. If $\mathcal{N}$ belongs to $\langle\!\langle \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$, then it follows by minimality of $\mathcal{M}$ that $\mathcal{M} = \mathcal{N}$. On the other hand, if $\mathcal{N}$ is of the form $\mathsf{aug}_\delta(\mathcal{N}_0, \langle\!\langle \mathcal{R} \rangle\!\rangle_{\mathsf{RE}})$ for some $\mathcal{N}_0$ from $\langle\!\langle \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$, then $\mathcal{N}_0 \subseteq \mathcal{N} \subseteq \mathcal{M}$, so by the minimality of $\mathcal{M}$, $\mathcal{N}_0 = \mathcal{N} = \mathcal{M}$. Thus, in either case, $\mathcal{M} = \mathcal{N}$, which proves that it is minimal within $\langle\!\langle \mathcal{R} \oplus \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$.

Now consider some δ_{d}- or δ_{e}-based rule update operator $\oplus$. Then, $\langle\!\langle \mathcal{R} \oplus \mathcal{R} \rangle\!\rangle_{\mathsf{RE}} \subseteq \langle\!\langle \mathcal{R} \rangle\!\rangle_{\mathsf{RE}} \cup \{\,\mathcal{X}\,\}$, so obviously $\langle\!\langle (\mathcal{R} \oplus \mathcal{R}) \cup \{\,\tau\,\} \rangle\!\rangle_{\mathsf{RE}} = \langle\!\langle \mathcal{R} \cup \{\,\tau\,\} \rangle\!\rangle_{\mathsf{RE}}$. $\square$

Remark 91 ((Idempotence) under $\equiv_{\mathsf{SR}}$ and $\equiv_{\mathsf{RR}}$)**.** The rule base $\mathcal{R} = \{\,(p \leftarrow \sim q.), (\sim p \leftarrow r.)\,\}$ forms a counterexample to (Idempotence) for δ_{a}-, δ_{b}- and δ_{c}-based operators with respect to SR and RR.

Lemma 92. *Let $\mathcal{M} \subseteq \mathcal{X}$ be RE-rule-expressible, $\mathcal{S}$ a set of RE-rule-expressible sets of three-valued interpretations. The following holds:*

   (i) If $\delta \in \{\,\delta_{\mathsf{b}}, \delta_{\mathsf{c}}\,\}$, then $\mathsf{aug}_\delta(\mathsf{aug}_\delta(\mathcal{M}, \mathcal{S}), \mathcal{S}) = \mathsf{aug}_\delta(\mathcal{M}, \mathcal{S})$.

   (ii) If $\delta \in \{\,\delta_{\mathsf{d}}, \delta_{\mathsf{e}}\,\}$, then either $\mathsf{aug}_\delta(\mathsf{aug}_\delta(\mathcal{M}, \mathcal{S}), \mathcal{S}) = \mathsf{aug}_\delta(\mathcal{M}, \mathcal{S})$, or both $\mathsf{aug}_\delta(\mathsf{aug}_\delta(\mathcal{M}, \mathcal{S}), \mathcal{S}) = \mathcal{X}$ and $\mathsf{aug}_\delta(\mathcal{M}, \mathcal{S}) \in \mathcal{S}$.

Proof. Let $\delta \in \{\,\delta_{\mathsf{b}}, \delta_{\mathsf{c}}\,\}$, put $\mathcal{M}' = \mathsf{aug}_\delta(\mathcal{M}, \mathcal{S})$ and take some $\mathcal{N} \in \mathcal{S}$ such that $\mathcal{M}' \bowtie^p_J \mathcal{N}$ for some atom p and some interpretation J. Then $\mathcal{M}'^J(p) = \mathsf{V}_0$ for some truth value V_0, so it follows from Lemma 75 that $\mathcal{M}^J(p) = \mathsf{V}_0$. But this implies $\mathcal{M} \bowtie^p_J \mathcal{N}$, so both $(J \setminus \{\,p\,\}, J \setminus \{\,p\,\})$ and $(J \cup \{\,p\,\}, J \cup \{\,p\,\})$ belong to $\mathcal{M}'$, a conflict with the assumption that $\mathcal{M}'^J(p)$ is defined. As a consequence, no such $\mathcal{N} \in \mathcal{S}$ exists, so $\mathsf{aug}_\delta(\mathcal{M}', \mathcal{S}) = \mathcal{M}' = \mathsf{aug}_\delta(\mathcal{M}, \mathcal{S})$.

On the other hand, if $\delta \in \{\,\delta_{\mathsf{d}}, \delta_{\mathsf{e}}\,\}$, then we can observe that either the previous case applies, or both $\mathsf{aug}_\delta(\mathsf{aug}_\delta(\mathcal{M}, \mathcal{S}), \mathcal{S}) = \mathcal{X}$ and $\mathsf{aug}_\delta(\mathcal{M}, \mathcal{S}) \in \mathcal{S}$. $\square$

Proposition 93. *Let $\oplus$ be a δ_{b}-, δ_{c}-, δ_{d}- or δ_{e}-based rule update operator. Then $\oplus$ satisfies* (Absorption) *with respect to RMR, SMR, RE, SE and SM. Moreover, if $\oplus$ is δ_{d}- or δ_{e}-based, then it satisfies* (Absorption) *with respect to RR and SR.*

Proof. (Absorption) states the following: $(\mathcal{R} \oplus \mathcal{U}) \oplus \mathcal{U} \equiv \mathcal{R} \oplus \mathcal{U}$. We will show that this is true under RMR-equivalence which, together with Proposition 15, implies that it holds under SMR-, RE-, SE- and SM-equivalence.

So suppose that $\oplus$ is δ_{b}-, δ_{c}-, δ_{d}- or δ_{e}-based. By Lemma 92, $\langle\!\langle ((\mathcal{R} \oplus \mathcal{U}) \oplus \mathcal{U}) \cup \{\,\tau\,\} \rangle\!\rangle_{\mathsf{RE}}$ is a superset of $\langle\!\langle (\mathcal{R} \oplus \mathcal{U}) \cup \{\,\tau\,\} \rangle\!\rangle_{\mathsf{RE}}$. Thus, whenever $\mathcal{M}$ is minimal in $\langle\!\langle ((\mathcal{R} \oplus \mathcal{U}) \oplus \mathcal{U}) \cup \{\,\tau\,\} \rangle\!\rangle_{\mathsf{RE}}$, it is also minimal in

$\langle\!\langle(\mathcal{R}\oplus\mathcal{U})\cup\{\tau\}\rangle\!\rangle_{\mathsf{RE}}$. Furthermore, the extra elements of $\langle\!\langle((\mathcal{R}\oplus\mathcal{U})\oplus\mathcal{U})\cup\{\tau\}\rangle\!\rangle_{\mathsf{RE}}$ are never smaller than the elements of $\langle\!\langle(\mathcal{R}\oplus\mathcal{U})\cup\{\tau\}\rangle\!\rangle_{\mathsf{RE}}$ because they are of the form $\mathsf{aug}_\delta(\mathcal{N},\langle\!\langle\mathcal{U}\rangle\!\rangle_{\mathsf{RE}})$ for some $\mathcal{N}\in\langle\!\langle\mathcal{U}\rangle\!\rangle_{\mathsf{RE}}\subseteq$ $\langle\!\langle(\mathcal{R}\oplus\mathcal{U})\cup\{\tau\}\rangle\!\rangle_{\mathsf{RE}}$. Thus, whenever $\mathcal{M}$ is minimal in $\langle\!\langle(\mathcal{R}\oplus\mathcal{U})\cup\{\tau\}\rangle\!\rangle_{\mathsf{RE}}$, it must also be minimal in $\langle\!\langle((\mathcal{R}\oplus\mathcal{U})\oplus\mathcal{U})\cup\{\tau\}\rangle\!\rangle_{\mathsf{RE}}$.

Furthermore, if $\oplus$ is δ_{d}- or δ_{e}-based, then it additionally holds that $\langle\!\langle((\mathcal{R}\oplus\mathcal{U})\oplus\mathcal{U})\cup\{\tau\}\rangle\!\rangle_{\mathsf{RE}}=$ $\langle\!\langle(\mathcal{R}\oplus\mathcal{U})\cup\{\tau\}\rangle\!\rangle_{\mathsf{RE}}$. $\square$

Proposition 94. *Let $\oplus$ be a δ_{b}-, δ_{c}-, δ_{d}- or δ_{e}-based rule update operator. Then $\oplus$ satisfies (Augmentation) for non-disjunctive programs with respect to RMR, SMR, RE, SE and SM. Moreover, if $\oplus$ is δ_{d}- or δ_{e}-based, then it satisfies (Augmentation) for non-disjunctive programs with respect to RR and SR.*

Proof. The proof for RMR-equivalence follows from Proposition 76 and Lemma 81 and from the fact that the extra elements of $\langle\!\langle((\mathcal{R}\oplus\mathcal{U})\oplus\mathcal{V})\cup\{\tau\}\rangle\!\rangle_{\mathsf{RE}}$, as compared to $\langle\!\langle(\mathcal{R}\oplus\mathcal{V})\cup\{\tau\}\rangle\!\rangle_{\mathsf{RE}}$, are non-minimal in the latter set. If $\oplus$ is δ_{d}- or δ_{e}-based, then there are no extra elements and the rest follows from Proposition 76 and Lemma 81. $\square$

Remark 95 ((Absorption) and (Augmentation) violated by δ_{a})**.** The rule bases $\mathcal{R}=\{p.\}$ and $\mathcal{U}=\mathcal{V}=$ $\{\sim p\leftarrow\sim q.,q.\}$ form counterexamples to (Absorption) and (Augmentation) for δ_{a}-based operators with respect SM-equivalence and any stronger notion of equivalence.

Remark 96 ((Augmentation) for Disjunctive Programs)**.** The following programs form a counterexample to (Augmentation) for δ_{b}-, δ_{c}-, δ_{d}- and δ_{e}-based operators under SM-equivalence (and thus under all stronger notions of equivalence as well):

$$\mathcal{P}:\quad p;q;r.\qquad\qquad \mathcal{U}:\quad\begin{aligned}&\sim p\leftarrow\sim r.\\&p\leftarrow r.\\&r\leftarrow p.\end{aligned}\qquad\qquad \mathcal{V}:\quad\begin{aligned}&\sim p\leftarrow\sim r.\\&p\leftarrow r.\\&r\leftarrow p.\\&\sim q.\end{aligned}$$

Remark 97 ((Associativity) and δ_{a}, δ_{b}, δ_{c}, δ_{d}, δ_{e})**.** The rule $\pi=(\sim p.)$, when updated by $\sigma=(p\leftarrow q.)$, must be weakened, anticipating the potential conflict. In the case of δ_{a}-, δ_{b}-, δ_{c}, δ_{d} and δ_{e}-based operators, the resulting rule is $\pi'=(\sim p\leftarrow\sim q.)$ (or another RE-equivalent rule). Consider the following rule bases:

$$\mathcal{R}=\{p.\},\quad\mathcal{U}=\{\sim p.\},\quad\mathcal{V}=\{p\leftarrow q.,q\leftarrow p.\}\ .$$

Note that (Associativity) states the following: $\mathcal{R}\oplus(\mathcal{U}\oplus\mathcal{V})\equiv(\mathcal{R}\oplus\mathcal{U})\oplus\mathcal{V}$. However, while in $(\mathcal{R}\oplus\mathcal{U})\oplus\mathcal{V}$ the fact from $\mathcal{R}$ is completely annihilated (i.e. transformed into a tautological rule) due to the negative fact π in $\mathcal{U}$, this does not happen in $\mathcal{R}\oplus(\mathcal{U}\oplus\mathcal{V})$ because the π is first weakened into π'. As a consequence, $\mathcal{R}\oplus(\mathcal{U}\oplus\mathcal{V})$ has one extra stable model comparing to $(\mathcal{R}\oplus\mathcal{U})\oplus\mathcal{V}$: $\{p,q\}$. This implies that (Associativity) does not hold for δ_{a}-, δ_{b}-, δ_{c}-, δ_{d}- and δ_{e}-based rule update operators under SM-equivalence, nor under any stronger equivalence.

Proposition 98. *Let $\oplus$ be a δ-based rule update operator. Then $\oplus$ satisfies (P2.1) and (P5) with respect to RMR, SMR, RE and SE.*

Proof. Under RMR-entailment (P2.1) follows from the fact that $\mathcal{M}$ is a subset of $\mathsf{aug}_\delta(\mathcal{M},\mathcal{S})$ and (P5) follows from the fact that $\mathsf{aug}_\delta(\mathcal{M},\mathcal{S})$ is a subset of $\mathsf{aug}_\delta(\mathcal{M},\mathcal{S}\cup\mathcal{T})$. For the remaining notions of program entailment this follows from Proposition 15. $\square$

Remark 99 ((P2.1) under $\models_{\mathsf{SR}}$ and $\models_{\mathsf{RR}}$). Consider again the rules π, σ from Remark 97 and rule bases $\mathcal{R} = \{\pi\}, \mathcal{U} = \{\sigma\}$. Note that (P2.1) states the following: $\mathcal{R} \cup \mathcal{U} \models \mathcal{R} \oplus \mathcal{U}$. However, if $\oplus$ is δ_{a}-, δ_{b}-, δ_{c}-, δ_{d}- or δ_{e}-based, $\mathcal{R} \oplus \mathcal{U}$ will contain π' (or another RE-equivalent rule or program) which results from weakening of π by σ. Consequently, when SR- or RR-entailment is used, $\mathcal{R} \cup \mathcal{U}$ cannot entail $\mathcal{R} \oplus \mathcal{U}$ simply because π' (or another RE-equivalent rule or program) does not belong to $\mathcal{R} \cup \mathcal{U}$.

Remark 100 ((P2.2) and Rule Updates). Consider $\mathcal{R} = \{p.\}$ and $\mathcal{U} = \{\sim p.\}$ and note that (P2.2) states the following: $(\mathcal{R} \cup \mathcal{U}) \oplus \mathcal{U} \models \mathcal{R}$. In other words, it requires that

$$\{p., \sim p.\} \oplus \{\sim p.\} \models p .$$

In the presence of (P1) this amounts to postulating that one can never recover from an inconsistent state. Such a requirement is out of line with the way these situations are treated in state-of-the-art approaches to rule updates which allow for recovery from an inconsistent state if all involved conflicts are resolved by the update. Note that, though for different reasons, (B2.2) has also been subject of harsh criticism in belief update literature [34].

Proposition 101. *Let $\oplus$ be a δ-based rule update operator where $\delta(\mathcal{M}, \mathcal{X}) \subseteq \mathcal{M}$ for all $\mathcal{M} \subseteq \mathcal{X}$. Then $\oplus$ satisfies (P4), (P4.1) and (P4.2) with respect to RR.*

Proof. Principle (P4) can be verified straightforwardly and (P4.1) as well as (P4.2) are its consequences. The condition on δ is necessary to ensure that $\mathsf{aug}_\delta(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}) = \mathcal{M}$ whenever $\mathcal{U}$ is tautological, to keep it in line with the case when $\mathcal{U} = \emptyset$. $\qquad\square$

Appendix C. Proofs: Condensing into a Nested Program

Lemma 102. *For any DLP $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$, $\bigoplus_{\mathsf{JU}} \boldsymbol{P}^\odot$ consists of the following rules:*

1. *for all $\pi \in \mathcal{P}_i^\odot$ with $i < n-1$, the nested rule $\left(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \bigwedge_{i<j<n} \sim \alpha_{\mathcal{P}_j^\odot}(\overline{\mathsf{H}_\pi}) . \right)$;*
2. *all nested rules in $\mathcal{P}_{n-1}^\odot$.*

Proof. Follows by induction on n. $\qquad\square$

Proposition 103. *Let I, J be interpretations and $\boldsymbol{P}$ a DLP. Then,*

$$I \models (\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}_{\mathsf{JU}}(\boldsymbol{P}, J))^J \qquad \text{if and only if} \qquad I \models \left(\bigoplus\nolimits_{\mathsf{JU}} \boldsymbol{P}^\odot \right)^J .$$

Proof. First suppose that $I \models (\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}_{\mathsf{JU}}(\boldsymbol{P}, J))^J$ and take some rule $\sigma^J \in (\bigoplus_{\mathsf{JU}} \boldsymbol{P}^\odot)^J$. We need to prove that $I \models \sigma^J$. It follows from Lemma 102 that one of the following cases occurs:

a) If $\sigma \in \mathcal{P}_{n-1}^\odot$, then it immediately follows that $\sigma \in (\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}_{\mathsf{JU}}(\boldsymbol{P}, J))^\odot$. Consequently, since $I \models (\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}_{\mathsf{JU}}(\boldsymbol{P}, J))^J$, it follows that $I \models \sigma^J$.

b) Otherwise, $\sigma = \left(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \bigwedge_{i<j<n} \sim \alpha_{\mathcal{P}_j^\odot}(\overline{\mathsf{H}_\pi}) . \right)$ for some $\pi \in \mathcal{P}_i^\odot$ with $i < n-1$. Suppose that $I \models \mathsf{B}_\sigma^J$. Then $I \models \mathsf{B}_\pi^J$ and it also follows that for all j with $i < j < n$, $J \not\models \alpha_{\mathcal{P}_j^\odot}(\overline{\mathsf{H}_\pi})$. Consequently, $\pi \in (\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}_{\mathsf{JU}}(\boldsymbol{P}, J))^\odot$ and by the assumption we conclude that $I \models \pi^J$. Hence, from $I \models \mathsf{B}_\pi^J$ it follows that $I \models \mathsf{H}_\pi^J$, implying that $I \models \sigma^J$.

For the converse implication, suppose that $I \models (\bigoplus_{\mathsf{JU}} \boldsymbol{P}^{\circledcirc})^J$ and take some $\pi \in (\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}_{\mathsf{JU}}(\boldsymbol{P}, J))^{\circledcirc}$. We need to show that $I \models \pi^J$. If $\pi \in \mathcal{P}_{n-1}^{\circledcirc}$, then it follows by Lemma 102 that $\pi \in (\bigoplus_{\mathsf{JU}} \boldsymbol{P}^{\circledcirc})$ and since $I \models (\bigoplus_{\mathsf{JU}} \boldsymbol{P}^{\circledcirc})^J$ by assumption, we can immediately conclude that $I \models \pi^J$.

In the principal case, $\pi \in \mathcal{P}_i^{\circledcirc}$ for some $i < n - 1$. Consequently, by Lemma 102, $\bigoplus_{\mathsf{JU}} \boldsymbol{P}^{\circledcirc}$ contains a rule $\sigma = \left(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \bigwedge_{i<j<n} \sim \alpha_{\mathcal{P}_j^{\circledcirc}}(\overline{\mathsf{H}_\pi}) . \right)$. Since π is not part of the set of rejected rules, we can conclude that for all j with $i < j < n$, $J \not\models \alpha_{\mathcal{P}_j^{\circledcirc}}(\overline{\mathsf{H}_\pi})$. Hence, $\mathsf{B}_\sigma^J = \mathsf{B}_\pi^J \wedge \bigwedge_{i<j<n} \top$. It follows that if $I \models \mathsf{B}_\pi^J$, then $I \models \mathsf{B}_\sigma^J$ and using our assumption we conclude that $I \models \mathsf{H}_\sigma^J$. Since $\mathsf{H}_\pi = \mathsf{H}_\sigma$, we demonstrated that $I \models \pi^J$. $\qquad \square$

Lemma 104. *For any DLP $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$, $\bigoplus_{\mathsf{UA}} \boldsymbol{P}^{\circledcirc}$ consists of the following rules:*

1. *for all $\pi \in \mathcal{P}_i^{\circledcirc}$ with $i < n - 1$, the nested rule $\left(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \bigwedge_{i<j<n} \sim \alpha_{\mathcal{P}_j^{\circledcirc}}(\overline{\mathsf{H}_\pi}) . \right)$;*

2. *for all $\pi \in \mathcal{P}_i^{\circledcirc}$ with $\mathsf{H}_\pi \in \mathcal{A}$ and $i < n$, the nested rule $\left(\mathsf{H}_\pi \vee \overline{\mathsf{H}_\pi} \leftarrow \mathsf{B}_\pi . \right)$;*

3. *all nested rules in $\mathcal{P}_{n-1}^{\circledcirc}$.*

Proof. Follows by induction on n. $\qquad \square$

Proposition 105. *Let $\boldsymbol{P}$ be a DLP and J an interpretation. If J is a UA-model of $\boldsymbol{P}$, then it is a stable model of $\bigoplus_{\mathsf{UA}} \boldsymbol{P}^{\circledcirc}$.*

Proof. In order to show that J is a stable model of $\bigoplus_{\mathsf{UA}} \boldsymbol{P}^{\circledcirc}$, we first establish that J satisfies $(\bigoplus_{\mathsf{UA}} \boldsymbol{P}^{\circledcirc})^J$. Take some rule σ from $\bigoplus_{\mathsf{UA}} \boldsymbol{P}^{\circledcirc}$. We need to prove that $J \models \sigma^J$. Due to Lemma 104, we need to consider the following cases:

1) In the first case, $\sigma = \left(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \bigwedge_{i<j<n} \sim \alpha_{\mathcal{P}_j^{\circledcirc}}(\overline{\mathsf{H}_\pi}) . \right)$ where $\pi \in \mathcal{P}_i^{\circledcirc}$ and $i < n - 1$. Suppose that $J \models \mathsf{B}_\sigma^J$. Then $J \models \mathsf{B}_\pi^J$ and for all j such that $i < j < n$, $J \not\models \alpha_{\mathcal{P}_j^{\circledcirc}}(\overline{\mathsf{H}_\pi})$. Thus, we can conclude that $\pi \in (\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}_{\mathsf{UA}}(\boldsymbol{P}, J))^{\circledcirc}$, by the assumption, $J \models \pi^J$. So since $J \models \mathsf{B}_\pi^J$, it also holds that $J \models \mathsf{H}_\pi^J$ and since $\mathsf{H}_\pi = \mathsf{H}_\sigma$, we conclude that $J \models \sigma^J$.

2) In the second case, $\sigma = \left(\mathsf{H}_\pi \vee \overline{\mathsf{H}_\pi} \leftarrow \mathsf{B}_\pi . \right)$ for some $\pi \in \mathcal{P}_i^{\circledcirc}$ with $\mathsf{H}_\pi \in \mathcal{A}$ and $i < n$. If $J \models \mathsf{H}_\pi$, then it easily follows that $J \models \sigma^J$. On the other hand, if $J \not\models \mathsf{H}_\pi$, then the head of σ^J contains $\top$ as the second disjunct and, once again, it follows that $J \models \sigma^J$.

3) In the third case, $\sigma \in \mathcal{P}_{n-1}^{\circledcirc}$. It immediately follows that $\sigma \in (\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}_{\mathsf{UA}}(\boldsymbol{P}, J))^{\circledcirc}$. Thus, our assumption implies that $J \models \sigma^J$.

It remains to verify that J is also subset-minimal among interpretations that satisfy $(\bigoplus_{\mathsf{UA}} \boldsymbol{P}^{\circledcirc})^J$. To show that this is the case, take an interpretation I with $I \subseteq J$ that satisfies $(\bigoplus_{\mathsf{UA}} \boldsymbol{P}^{\circledcirc})^J$. In the following we will prove that $I \models (\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}_{\mathsf{UA}}(\boldsymbol{P}, J))^J$. Since J is subset-minimal among interpretations satisfying this program, it will follow that $I = J$ as desired.

Take some rule $\pi \in (\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}_{\mathsf{UA}}(\boldsymbol{P}, J))^{\circledcirc}$, our goal is to prove that $I \models \pi^J$. We consider the following cases:

1) In case $\pi \in \mathcal{P}_i^{\circledcirc}$ with $i < n - 1$ and there is no rule $\pi' \in \mathcal{P}_j$ with $i < j < n$ such that $\pi' \bowtie \pi$ and $J \models \mathsf{B}_{\pi'}$, we can use Lemma 104 to conclude that $\bigoplus_{\mathsf{UA}} \boldsymbol{P}^{\circledcirc}$ contains a rule $\sigma = \left(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \bigwedge_{i<j<n} \sim \alpha_{\mathcal{P}_j^{\circledcirc}}(\overline{\mathsf{H}_\pi}) . \right)$. It also follows that for all j with $i < j < n$, $J \not\models \alpha_{\mathcal{P}_j^{\circledcirc}}(\overline{\mathsf{H}_\pi})$, so that $\mathsf{B}_\sigma^J = \mathsf{B}_\pi^J \wedge \bigwedge_{i<j<n} \top$. Thus, if $I \models \mathsf{B}_\pi^J$, then $I \models \mathsf{B}_\sigma^J$ and by the assumption that $I \models (\bigoplus_{\mathsf{UA}} \boldsymbol{P}^{\circledcirc})^J$ we obtain $I \models \mathsf{H}_\sigma^J$, so it also follows that $I \models \pi^J$.

2) In case $\pi \in \mathcal{P}_i^{\circledcirc}$ with $i < n - 1$ and there is a rule $\pi' \in \mathcal{P}_j$ with $i < j < n$ such that $\pi' \bowtie \pi$ and $J \models \mathsf{B}_{\pi'}$, it follows that since π is unrejected, π' is itself rejected. Consequently, there is also an unrejected rule $\pi'' \in \mathcal{P}_k$ with $j < k < n$ such that $\pi'' \bowtie \pi'$ and $J \models \mathsf{B}_{\pi''}$. Furthermore, $\mathsf{H}_{\pi''} = \mathsf{H}_\pi$ and by the assumption we know that J satisfies the rule π'', so we can conclude that $J \models \mathsf{H}_\pi$. If H_π is a default literal, then it follows that $(\mathsf{H}_\pi)^J = \top$, so trivially $I \models \pi^J$.
   If H_π is an atom, then, by Lemma 104, $\bigoplus_{\mathsf{UA}} \boldsymbol{P}^\circledcirc$ contains a rule $\sigma = \left(\mathsf{H}_\pi \vee \overline{\mathsf{H}_\pi} \leftarrow \mathsf{B}_\pi.\right)$. Note that $\mathsf{H}_\sigma^J = \mathsf{H}_\pi \vee \bot$. Thus, if $I \models \mathsf{B}_\pi^J$, then $I \models \mathsf{B}_\sigma^J$ and it follows from our assumption that $I \models \mathsf{H}_\sigma^J$, so that $I \models \mathsf{H}_\pi^J$. Hence, $I \models \pi^J$.

3) In case $\pi \in \mathcal{P}_{n-1}^{\circledcirc}$, it immediately follows from Lemma 104 that $\pi \in \left(\bigoplus_{\mathsf{UA}} \boldsymbol{P}^\circledcirc\right)$. Thus, by the assumption that $I \models \left(\bigoplus_{\mathsf{UA}} \boldsymbol{P}^\circledcirc\right)^J$ we obtain $I \models \pi^J$. $\qquad\square$

Proposition 106. *Let $\boldsymbol{P}$ be a DLP and J an interpretation. If J is a stable model of $\bigoplus_{\mathsf{UA}} \boldsymbol{P}^\circledcirc$, then it is a UA-model of $\boldsymbol{P}$.*

Proof. To show that J is a UA-model of $\boldsymbol{P}$, we first establish that J satisfies $\mathcal{P} = \left(\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}_{\mathsf{UA}}(\boldsymbol{P}, J)\right)^J$. Take some rule $\pi \in \left(\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}_{\mathsf{UA}}(\boldsymbol{P}, J)\right)^\circledcirc$, our goal is to prove that $J \models \pi^J$. We consider the following cases:

1) In case $\pi \in \mathcal{P}_i^{\circledcirc}$ with $i < n - 1$ and there is no rule $\pi' \in \mathcal{P}_j$ with $i < j < n$ such that $\pi' \bowtie \pi$ and $J \models \mathsf{B}_{\pi'}$, we can use Lemma 104 to conclude that $\bigoplus_{\mathsf{UA}} \boldsymbol{P}^\circledcirc$ contains a rule $\sigma = \left(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \bigwedge_{i < j < n} \sim \alpha_{\mathcal{P}_j^{\circledcirc}}(\overline{\mathsf{H}_\pi}).\right)$. It also follows that for all j with $i < j < n$, $J \not\models \alpha_{\mathcal{P}_j^{\circledcirc}}(\overline{\mathsf{H}_\pi})$, so that $\mathsf{B}_\sigma^J = \mathsf{B}_\pi^J \wedge \bigwedge_{i < j < n} \top$. Thus, if $J \models \mathsf{B}_\pi^J$, then $J \models \mathsf{B}_\sigma^J$ and by the assumption that $J \models \left(\bigoplus_{\mathsf{UA}} \boldsymbol{P}^\circledcirc\right)^J$ we obtain $J \models \mathsf{H}_\sigma^J$, so it follows that $J \models \pi^J$.

2) In case $\pi \in \mathcal{P}_i^{\circledcirc}$ with $i < n - 1$ and there is a rule $\pi' \in \mathcal{P}_j$ with $i < j < n$ such that $\pi' \bowtie \pi$ and $J \models \mathsf{B}_{\pi'}$, it follows that since π is unrejected, π' is itself rejected. Take the maximal index k such that $\mathcal{P}_k$ contains a rule π'' with $\pi'' \bowtie \pi'$ and $J \models \mathsf{H}_{\pi''}$. It follows that π'' satisfies the condition of the previous case, and thus $J \models \mathsf{H}_{\pi''}^J$. Since $\mathsf{H}_\pi = \mathsf{H}_{\pi''}$, we conclude that $J \models \pi^J$.

3) In case $\pi \in \mathcal{P}_{n-1}^{\circledcirc}$, it immediately follows from Lemma 104 that $\pi \in \left(\bigoplus_{\mathsf{UA}} \boldsymbol{P}^\circledcirc\right)$. Thus, by the assumption that $J \models \left(\bigoplus_{\mathsf{UA}} \boldsymbol{P}^\circledcirc\right)^J$ we obtain that $J \models \pi^J$.

It remains to verify that J is also subset-minimal among interpretations that satisfy $\mathcal{P}$. To show that this is the case, take an interpretation I with $I \subseteq J$ that satisfies $\mathcal{P}$. In the following we will prove that I also satisfies the program $\left(\bigoplus_{\mathsf{UA}} \boldsymbol{P}^\circledcirc\right)^J$. Since J is subset-minimal among interpretations satisfying this program, it will follow that $I = J$ as desired.

So take some $\sigma \in \left(\bigoplus_{\mathsf{UA}} \boldsymbol{P}^\circledcirc\right)$, our goal is to prove that $I \models \sigma^J$. Due to Lemma 104, we need to consider the following cases:

1) In the first case, $\sigma = \left(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \bigwedge_{i < j < n} \sim \alpha_{\mathcal{P}_j^{\circledcirc}}(\overline{\mathsf{H}_\pi}).\right)$ where $\pi \in \mathcal{P}_i^{\circledcirc}$ and $i < n - 1$. Suppose that $I \models \mathsf{B}_\sigma^J$. Then $I \models \mathsf{B}_\pi^J$ and for all j such that $i < j < n$, $J \not\models \alpha_{\mathcal{P}_j^{\circledcirc}}(\overline{\mathsf{H}_\pi})$. Thus, we can conclude that $\pi \in \left(\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}_{\mathsf{UA}}(\boldsymbol{P}, J)\right)^\circledcirc$. Furthermore, from $I \models \mathsf{B}_\pi^J$ and the assumption that $I \models \mathcal{P}$, it follows that $I \models \mathsf{H}_\pi^J$. Consequently, since $\mathsf{H}_\pi = \mathsf{H}_\sigma$, $I \models \sigma^J$.

2) In the second case, $\sigma = (\mathsf{H}_\pi \vee \overline{\mathsf{H}_\pi} \leftarrow \mathsf{B}_\pi.)$ for some $\pi \in \mathcal{P}_i^{\circledcirc}$ with $\mathsf{H}_\pi \in \mathcal{A}$ and $i < n$. If π is not rejected, then it follows from our assumption that $I \models \pi^J$. In the principal case, there exists a rule $\pi' \in \mathcal{P}_j$ with $i < j < n$ such that $\pi' \bowtie \pi$ and $J \models \mathsf{B}_{\pi'}$. Furthermore, π' itself is not rejected, so due to our previous considerations we can conclude that $J \models \mathsf{H}_{\pi'}$. Note that since H_π is an atom, $\mathsf{H}_{\pi'}$ is a default literal, so the rule σ^J has $\top$ as one of the disjuncts in its head. Thus, I trivially satisfies σ^J.

3) In the third case, $\sigma \in \mathcal{P}_{n-1}^{\odot}$. It immediately follows that $\sigma \in (\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}_{\mathsf{UA}}(\boldsymbol{P}, J))^{\odot}$. Thus, our assumption implies that $I \models \sigma^J$. $\qquad\square$

Proof of Theorem 28. Follows from Propositions 103, 105 and 106. $\qquad\square$

Appendix D. Proofs: Condensing into a Disjunctive Program

Remark 107. Throughout the following proofs we abuse notation by ignoring differences between formulas that can be eliminated by regrouping and reordering conjuncts and disjuncts within them. That is, when a formula can be obtained from another formula only by using the commutative and associative laws for conjunction and disjunction, we consider the two formulas identical. We can afford to do this because the order and grouping of conjuncts and disjuncts has no effect on the resulting semantics.

Definition 108 (Strong Equivalence [50]). Let $\mathcal{P}$, $\mathcal{Q}$ be programs. We say that $\mathcal{P}$ *is strongly equivalent to* $\mathcal{Q}$ if for every program $\mathcal{R}$, the stable models of $\mathcal{P} \cup \mathcal{R}$ coincide with the stable models of $\mathcal{Q} \cup \mathcal{R}$. Strong equivalence is extended to rules by treating each rule π as the program $\{\,\pi\,\}$.

Proposition 109 ([50, 71]). *Let $\mathcal{P}$, $\mathcal{Q}$ be programs and π a rule. If for all interpretations I, J with $I \subseteq J$, $J \models \mathcal{P} \wedge I \models \mathcal{P}^J$ if and only if $J \models \mathcal{Q} \wedge I \models \mathcal{Q}^J$, then $\mathcal{P}$ is strongly equivalent to $\mathcal{Q}$.*

Proof. Take some program $\mathcal{R}$ and some interpretation J. J is a stable model of $\mathcal{P} \cup \mathcal{R}$ if and only if $J \models (\mathcal{P} \cup \mathcal{R})^J$ and $\forall I \subsetneq J : I \not\models (\mathcal{P} \cup \mathcal{R})^J$. Due to the definition of reduct and the assumption, this is equivalent to $J \models (\mathcal{Q} \cup \mathcal{R})^J$ and $\forall I \subsetneq J : I \not\models (\mathcal{Q} \cup \mathcal{R})^J$. In other words, J is a stable model of $\mathcal{P} \cup \mathcal{R}$ if and only if it is a stable model of $\mathcal{Q} \cup \mathcal{R}$. $\qquad\square$

Corollary 110. *Let $\mathcal{P}$, $\mathcal{Q}$ be programs and π a rule. If for all interpretations I, J, $I \models \mathcal{P}^J$ if and only if $I \models \mathcal{Q}^J$, then $\mathcal{P}$ is strongly equivalent to $\mathcal{Q}$.*

Proof. Follows from Proposition 109 and the fact that for every interpretation K and program $\mathcal{R}$, $K \models \mathcal{R}$ if and only if $K \models \mathcal{R}^K$. $\qquad\square$

Corollary 111. *Let ϕ, ψ be formulas and π a rule. If for all interpretations I, J, $I \models \phi^J$ if and only if $I \models \psi^J$, then the rules $(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \phi.)$ and $(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \psi.)$ are strongly equivalent.*

Proof. Follows from the definition of rule reduct and from Corollary 110. $\qquad\square$

Definition 112. Given a set of formulas S, we define $\sim S = \{\,\sim\phi \mid \phi \in S\,\}$.

Lemma 113. *Let $\mathcal{P}$ be a program, L a literal, S be a set of formulas and I, J interpretations. Then,*

(i) $I \models (\bigwedge \sim S)^J$ *if and only if* $I \models (\sim \bigvee S)^J$;

(ii) $I \models (\sim \alpha_{\mathcal{P}^{\odot}}(L))^J$ *if and only if* $I \models \left(\bigvee_{S \in \beta_{\mathcal{P}}(L)} \bigwedge \sim\sim S^+ \wedge \bigwedge \sim S^- \right)^J$.

Proof.

(i) By the definition of reduct, $I \models (\bigwedge \sim S)^J$ if and only if for all $\phi \in S$ it holds that $J \not\models \phi$. This in turn holds if and only if $J \not\models \bigvee S$, which is the case if and only if $I \models (\sim \bigvee S)^J$.

55

(ii) Suppose that $\alpha_{\mathcal{P}\circledcirc}(L)$ is the formula $(L_1^1 \wedge \cdots \wedge L_{k_1}^1) \vee \cdots \vee (L_1^n \wedge \cdots \wedge L_{k_n}^n)$. By the definition of reduct, $I \models (\sim\alpha_{\mathcal{P}\circledcirc}(L))^J$ if and only if $J \not\models \alpha_{\mathcal{P}\circledcirc}(L)$. Equivalently, for every j with $1 \leq j \leq n$ there exists some i_j with $1 \leq i_j \leq k_j$ such that $J \not\models L_{i_j}^j$. By the definition of blocking sets, this is equivalent to $J \models S$ for some $S \in \beta_{\mathcal{P}}(L)$. Equivalently, for some $S \in \beta_{\mathcal{P}}(L)$,

$$\forall L \in \sim S^+ : J \not\models L \qquad\qquad \text{and} \qquad\qquad \forall L \in S^- : J \not\models L \ ,$$

or in other words, $(\bigwedge \sim\sim S^+ \wedge \bigwedge \sim S^-)^J = \bigwedge_{L \in S} \top$. Equivalently, we can also write $I \models$ $\left(\bigvee_{S \in \beta_{\mathcal{P}}(L)} \bigwedge \sim\sim S^+ \wedge \bigwedge \sim S^-\right)^J$. $\qquad\square$

Lemma 114. *Let π be a rule and S a set of formulas. Then the rule $(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \bigvee S.)$ is strongly equivalent to the program $\{\, \mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \phi \mid \phi \in S \,\}$.*

Proof. Let σ denote the rule and $\mathcal{P}$ the program. According to Corollary 110, it suffices to prove that for all interpretations I, J, it holds that $I \models \sigma^J$ if and only if it holds that $I \models \mathcal{P}^J$. This easily follows from the fact that $I \models \sigma^J$ if and only if $I \models \mathsf{B}_\sigma^J$ implies $I \models \mathsf{H}_\sigma^J$, or, equivalently, for all $\phi \in S$, $I \models \mathsf{B}_\pi^J \wedge \phi^J$ implies $I \models \mathsf{H}_\pi^J$, which is another way of writing $I \models \mathcal{P}^J$. $\qquad\square$

Lemma 115. *Let π be a rule and ϕ a formula. Then the rules $(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \sim\sim\phi.)$ and $(\mathsf{H}_\pi \vee \sim\phi \leftarrow \mathsf{B}_\pi.)$ are strongly equivalent.*

Proof. Let σ_1 denote the first rule and σ_2 the second. By Corollary 110, it suffices to prove that for all interpretations I, J, it holds that $I \models \sigma_1^J$ if and only if it holds that $I \models \sigma_2^J$. If $J \models \phi$, then $\sigma_1^J = (\mathsf{H}_\pi^J \leftarrow \mathsf{B}_\pi^J \wedge \top)$ and $\sigma_2^J = (\mathsf{H}_\pi^J \vee \bot \leftarrow \mathsf{B}_\pi^J)$, so it follows that $I \models \sigma_1^J$ holds if and only if $I \models \sigma_2^J$.

On the other hand, if $J \not\models \phi$, then $\sigma_1^J = (\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \bot)$ $\sigma_2^J = (\mathsf{H}_\pi \vee \top \leftarrow \mathsf{B}_\pi)$ and trivially both $I \models \sigma_1^J$ and $I \models \sigma_2^J$ hold. $\qquad\square$

Lemma 116. *Let π be a rule such that H_π is a default literal and B_π is a conjunction of literals and double-negated atoms. For any atom p, the rules $(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \sim\sim p.)$ and $(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge p.)$ are are strongly equivalent.*

Proof. Let $\mathsf{H}_\pi = \sim q$, where q is an atom. Also, let σ_1 denote the first rule and σ_2 the second. By Proposition 109, it suffices to prove that for all interpretations I, J with $I \subseteq J$,

$$J \models \sigma_1 \wedge I \models \sigma_1^J \qquad\qquad \text{if and only if} \qquad\qquad J \models \sigma_2 \wedge I \models \sigma_2^J \ ,$$

First suppose that $J \models \sigma_1$ and $I \models \sigma_1^J$. Then clearly $J \models \sigma_2$ and it remains to prove that $I \models \sigma_2^J$. Suppose that $I \models \mathsf{B}_{\sigma_2}^J$. Then $I \models \mathsf{B}_\pi^J$ and from $I \subseteq J$ we conclude that $J \models p$, so $I \models (\sim\sim p)^J$. Consequently, $I \models \mathsf{B}_{\sigma_1}^J$ and from the assumption that $I \models \sigma_1^J$ we conclude that $I \models \mathsf{H}_\pi^J$ as desired.

Now suppose that $J \models \sigma_2$ and $I \models \sigma_2^J$. Then clearly $J \models \sigma_1$ and it remains to prove that $I \models \sigma_1^J$. Suppose that $I \models \mathsf{B}_{\sigma_1}^J$. Then $I \models \mathsf{B}_\pi^J$, so it follows from $I \subseteq J$ that $J \models \mathsf{B}_\pi$. Furthermore, from $I \models (\sim\sim p)^J$ it follows that $J \models p$. Thus, $J \models \mathsf{B}_{\sigma_2}$ and from $J \models \sigma_2$ we conclude that $J \models \mathsf{H}_\pi$. This implies that $\mathsf{H}_\pi^J = (\sim q)^J = \top$ and, thus, $I \models \mathsf{H}_\pi^J$ as desired. $\qquad\square$

Lemma 117. *Let π be a rule and ϕ a formula such that H_π is an atom and both B_π and ϕ are conjunctions of literals and double-negated atoms. For any atom p, the programs $\{\, \mathsf{H}_\pi \vee \overline{\mathsf{H}_\pi} \leftarrow \mathsf{B}_\pi;\ \mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \phi \wedge \sim\sim p \,\}$ and $\{\, \mathsf{H}_\pi \vee \overline{\mathsf{H}_\pi} \leftarrow \mathsf{B}_\pi;\ \mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \phi \wedge p \,\}$ are strongly equivalent.*

Proof. Let $\mathcal{P}_1$ denote the first program and $\mathcal{P}_2$ the second one. Also, let σ denote the rule $(\mathsf{H}_\pi \vee \overline{\mathsf{H}_\pi} \leftarrow \mathsf{B}_\pi)$, σ_1 the rule $(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \phi \wedge \sim\sim p)$ and σ_2 the rule $(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \phi \wedge p)$. Thus, $\mathcal{P}_1 = \{\,\sigma, \sigma_1\,\}$ and $\mathcal{P}_2 = \{\,\sigma, \sigma_2\,\}$.

According to Proposition 109, it suffices to prove that for all interpretations I, J with $I \subseteq J$,

$$J \models \mathcal{P}_1 \wedge I \models \mathcal{P}_1^J \qquad\qquad \text{if and only if} \qquad\qquad J \models \mathcal{P}_2 \wedge I \models \mathcal{P}_2^J \ ,$$

First assume that $J \models \mathcal{P}_1$ and $I \models \mathcal{P}_1^J$. Clearly, it follows that $J \models \mathcal{P}_2$ and $I \models \sigma^J$, so it remains to verify that $I \models \sigma_2^J$. Suppose that $I \models \mathsf{B}_{\sigma_2}^J$. Then, since $I \subseteq J$, it follows that $J \models p$ and so $(\sim\sim p)^J = \top$. Consequently, $I \models \mathsf{B}_{\sigma_1}^J$ and from the assumption that $I \models \mathcal{P}_1^J$ we can conclude that $I \models \mathsf{H}_\pi^J$ as desired.

For the converse implication, assume that $J \models \mathcal{P}_2$ and $I \models \mathcal{P}_2^J$. It immediately follows that $J \models \mathcal{P}_1$ and $I \models \sigma^J$, so it remains to verify that $I \models \sigma_1^J$. Suppose that $I \models \mathsf{B}_{\sigma_1}^J$. From $I \subseteq J$ and the assumption that both B_π and ϕ are conjunctions of literals and double-negated atoms, we conclude that $J \models \mathsf{B}_{\sigma_1}$, so it follows that $J \models \mathsf{H}_\pi$. Consequently, $(\overline{\mathsf{H}_\pi})^J = \bot$ and since $I \models \sigma^J$, $I \models \mathsf{B}_\pi^J$ implies that $I \models \mathsf{H}_\pi^J$ as desired. $\qquad\square$

Lemma 118. *Let $\mathcal{P}$, $\mathcal{Q}$ be programs and L a literal. Then,*

$$\beta_{\mathcal{P}\cup\mathcal{Q}}(L) = \{\, S \cup T \mid S \in \beta_{\mathcal{P}}(L) \wedge T \in \beta_{\mathcal{Q}}(L)\,\} \ .$$

Proof. Suppose that $\alpha_{\mathcal{P}^\circledcirc}(L)$, $\alpha_{\mathcal{Q}^\circledcirc}(L)$ and $\alpha_{(\mathcal{P}\cup\mathcal{Q})^\circledcirc}(L)$ are, respectively, of the forms

$$(L_1^1 \wedge \cdots \wedge L_{k_1}^1) \vee \cdots \vee (L_1^m \wedge \cdots \wedge L_{k_m}^m) \ ,$$
$$(L_1^{m+1} \wedge \cdots \wedge L_{k_{m+1}}^{m+1}) \vee \cdots \vee (L_1^n \wedge \cdots \wedge L_{k_n}^n) \ ,$$
$$(L_1^1 \wedge \cdots \wedge L_{k_1}^1) \vee \cdots \vee (L_1^n \wedge \cdots \wedge L_{k_n}^n) \ .$$

By the definition, $S \in \beta_{\mathcal{P}}(L)$ and $T \in \beta_{\mathcal{Q}}(L)$ if and only if

$$S = \left\{\, \overline{L_{i_1}^1}, \ldots, \overline{L_{i_m}^m}\,\right\} \qquad\qquad \text{and} \qquad\qquad T = \left\{\, \overline{L_{i_{m+1}}^{m+1}}, \ldots, \overline{L_{i_n}^n}\,\right\}$$

where $1 \le i_j \le k_j$ for every j with $1 \le j \le n$. This is equivalent to $S \cup T$ being a member of $\beta_{\mathcal{P}\cup\mathcal{Q}}(L)$. $\qquad\square$

Lemma 119. *Let $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ be a DLP. Then $\bigoplus_{\mathsf{JU}}^\vee \boldsymbol{P}$ consists of the following rules:*

1. *for all $\pi \in \mathcal{P}_i$ with $i < n-1$ such that $\mathsf{H}_\pi = \{\,p\,\}$ for some $p \in \mathcal{A}$, and all $S \in \beta_{\bigcup_{i<j<n}\mathcal{P}_j}(\sim p)$, the rule $\left(\mathsf{H}_\pi; \overline{S^+} \leftarrow \mathsf{B}_\pi, \overline{S^-}.\right)$;*

2. *for all $\pi \in \mathcal{P}_i$ with $i < n-1$ such that $\mathsf{H}_\pi = \{\sim p\,\}$ for some $p \in \mathcal{A}$, and all $S \in \beta_{\bigcup_{i<j<n}\mathcal{P}_j}(p)$, the rule $\left(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi, S^+, \overline{S^-}.\right)$;;*

3. *all rules in $\mathcal{P}_{n-1}$.*

Proof. Follows by induction on n using Lemma 118. $\qquad\square$

Lemma 120. *Let $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ be a DLP. Then $\bigoplus_{\mathsf{UA}}^\vee \boldsymbol{P}$ consists of the following rules:*

1. *for all $\pi \in \mathcal{P}_i$ with $i < n-1$ and all $S \in \beta_{\bigcup_{i<j<n}\mathcal{P}_j}(\overline{\mathsf{H}_\pi})$, the rule $\left(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi, S^+, \overline{S^-}.\right)$;*

2. *for all $\pi \in \mathcal{P}_i$ with $i < n$ such that $\mathsf{H}_\pi = \{\,p\,\}$ for some $p \in \mathcal{A}$, the rule $(p; \sim p \leftarrow \mathsf{B}_\pi.)$;*

3. *all rules in $\mathcal{P}_{n-1}$.*

Proof. Follows by induction on n using Lemma 118. □

Proof of Theorem 31.

(i) Due to Theorem 28, it suffices to show that the programs $\bigoplus_{\mathsf{JU}} \boldsymbol{P}^{\circledcirc}$ and $\bigoplus_{\mathsf{JU}}^{\vee} \boldsymbol{P}$ have the same stable models. To see that this is indeed the case, consider the contents of these programs, as established in Lemmas 102 and 119. For all $\pi \in \mathcal{P}_i^{\circledcirc}$ with $i < n - 1$, $\bigoplus_{\mathsf{JU}} \boldsymbol{P}^{\circledcirc}$ contains the nested rule $\left(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \bigwedge_{i<j<n} \sim \alpha_{\mathcal{P}_j^{\circledcirc}}(\overline{\mathsf{H}_\pi}).\right)$. By Lemma 113(i) and Corollary 111, this rule is strongly equivalent to the rule $\left(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \sim \bigvee_{i<j<n} \alpha_{\mathcal{P}_j^{\circledcirc}}(\overline{\mathsf{H}_\pi}).\right)$ which, by the definition of activation condition, can also be written as $\left(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \sim \alpha_{\bigcup_{i<j<n} \mathcal{P}_j^{\circledcirc}}(\overline{\mathsf{H}_\pi}).\right)$. Furthermore, due to Lemma 113(ii) and Corollary 111, the latter rule is strongly equivalent to the rule

$$\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \bigvee_{S \in \beta_{\bigcup_{i<j<n} \mathcal{P}_j}(\overline{\mathsf{H}_\pi})} \bigwedge \sim \sim S^+ \wedge \bigwedge \sim S^-.$$

and by using Lemma 114 we obtain the strongly equivalent program that contains, for each $S \in \beta_{\bigcup_{i<j<n} \mathcal{P}_j}(\overline{\mathsf{H}_\pi})$, the rule

$$\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \bigwedge \sim \sim S^+ \wedge \bigwedge \sim S^-.$$

Double-negated atoms from the bodies of these rules can be eliminated using Lemmas 115 and 116, obtaining $(\mathsf{H}_\pi \vee \bigvee \sim S^+ \leftarrow \mathsf{B}_\pi \wedge \bigwedge \sim S^-.)$ if H_π is an atom and $(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \bigwedge S^+ \wedge \bigwedge \sim S^-.)$ if H_π is a default literal. In this way, the original nested rules can be converted, one at a time, into a strongly equivalent disjunctive program. After this process is finished, the nested syntax can be converted to the syntax of disjunctive programs and the result coincides with the disjunctive program $\bigoplus_{\mathsf{JU}}^{\vee} \boldsymbol{P}$.

(ii) Due to Theorem 28, it suffices to show that the programs $\bigoplus_{\mathsf{UA}} \boldsymbol{P}^{\circledcirc}$ and $\bigoplus_{\mathsf{UA}}^{\vee} \boldsymbol{P}$ have the same stable models. To see that this is indeed the case, consider the contents of these programs, as established in Lemmas 104 and 120. For all $\pi \in \mathcal{P}_i^{\circledcirc}$ with $i < n - 1$, $\bigoplus_{\mathsf{UA}} \boldsymbol{P}$ contains the nested rule $\left(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \bigwedge_{i<j<n} \sim \alpha_{\mathcal{P}_j^{\circledcirc}}(\overline{\mathsf{H}_\pi}).\right)$. By Lemma 113(i) and Corollary 111, this rule is strongly equivalent to the rule $\left(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \sim \bigvee_{i<j<n} \alpha_{\mathcal{P}_j^{\circledcirc}}(\overline{\mathsf{H}_\pi}).\right)$ which, by the definition of activation condition, can also be written as $\left(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \sim \alpha_{\bigcup_{i<j<n} \mathcal{P}_j^{\circledcirc}}(\overline{\mathsf{H}_\pi}).\right)$. Furthermore, due to Lemma 113(ii) and Corollary 111, the latter rule is strongly equivalent to the rule

$$\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \bigvee_{S \in \beta_{\bigcup_{i<j<n} \mathcal{P}_j}(\overline{\mathsf{H}_\pi})} \bigwedge \sim \sim S^+ \wedge \bigwedge \sim S^-.$$

and by using Lemma 114 we obtain the strongly equivalent program that contains, for each $S \in \beta_{\bigcup_{i<j<n} \mathcal{P}_j}(\overline{\mathsf{H}_\pi})$, the rule

$$\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \bigwedge \sim \sim S^+ \wedge \bigwedge \sim S^-.$$

Finally, double-negated atoms from the bodies of these rules can be eliminated using Lemmas 117 and 116, obtaining $(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi \wedge \bigwedge S^+ \wedge \bigwedge \sim S^-.)$ In this way, the original nested rules can be converted, one at a time, into a strongly equivalent disjunctive program. After this process is finished, the nested syntax can be converted to the syntax of disjunctive programs and the result coincides with the disjunctive program $\bigoplus_{\mathsf{UA}}^{\vee} \boldsymbol{P}$. □

Appendix E. Proofs: Belief Updates Using Exception-Based Operators

Appendix E.1. Model-Based Update Operators

Theorem 34. *If $\diamond$ is an update operator that satisfies (B1), (B2.1) and (B4), then there exists an exception function ε such that for every ε-based update operator $\oplus$ and all finite sequences of knowledge bases $\mathcal{D}$, $[\![\diamond\,\mathcal{D}]\!] = [\![\bigoplus\mathcal{D}]\!]$.*

Proof. Let the exception function ε be defined for all sets of interpretations $\mathcal{M} \subseteq \mathfrak{I}$ and all sets of sets of interpretations $\mathcal{S}, \mathcal{T} \subseteq 2^{\mathfrak{I}}$ as

$$\varepsilon(\mathcal{M}, \mathcal{S}, \mathcal{T}) = [\![\mathcal{B}\diamond\mathcal{U}]\!] \tag{E.1}$$

where $\mathcal{B}, \mathcal{U}$ are some knowledge bases such that $[\![\mathcal{B}]\!] = \bigcap\mathcal{S}$ and $[\![\mathcal{U}]\!] = \bigcap\mathcal{T}$. Note that this definition is unambiguous since the existence of such $\mathcal{B}$ and $\mathcal{U}$ is guaranteed and regardless of which pair of knowledge bases with these properties we choose, we obtain the same result due to the assumption that $\diamond$ satisfies (B4). Take some ε-based operator $\oplus$. We proceed by induction on the length n of $\mathcal{D} = \langle\mathcal{B}_i\rangle_{i<n}$.

1° If $n = 0$, then it immediately follows that $[\![\diamond\,\mathcal{D}]\!] = [\![\diamond\langle\mathcal{B}_0\rangle]\!] = [\![\mathcal{B}_0]\!] = [\![\bigoplus\langle\mathcal{B}_0\rangle]\!] = [\![\bigoplus\mathcal{D}]\!]$.

2° Suppose that the claim holds for n, i.e. for $\mathcal{D} = \langle\mathcal{B}_i\rangle_{i<n}$ we have $[\![\diamond\,\mathcal{D}]\!] = [\![\bigoplus\mathcal{D}]\!]$. Our goal is to show that it also holds for $n+1$, i.e. for $\mathcal{D}' = \langle\mathcal{B}_i\rangle_{i<n+1}$. It follows that

$$\langle\!\langle\bigoplus\mathcal{D}'\rangle\!\rangle = \langle\!\langle\bigoplus\mathcal{D}\oplus\mathcal{B}_n\rangle\!\rangle = \{\,[\![\phi]\!]\cup\varepsilon([\![\phi]\!], \langle\!\langle\bigoplus\mathcal{D}\rangle\!\rangle, \langle\!\langle\mathcal{B}_n\rangle\!\rangle) \mid \phi\in\bigoplus\mathcal{D}\,\}\cup\langle\!\langle\mathcal{B}_n\rangle\!\rangle\ .$$

By (E.1) and the inductive assumption, $\varepsilon([\![\phi]\!], \langle\!\langle\bigoplus\mathcal{D}\rangle\!\rangle, \langle\!\langle\mathcal{B}_n\rangle\!\rangle) = [\![\diamond\,\mathcal{D}\diamond\mathcal{B}_n]\!] = [\![\diamond\,\mathcal{D}']\!]$. Consequently,

$$[\![\bigoplus\mathcal{D}']\!] = \bigcap\langle\!\langle\bigoplus\mathcal{D}'\rangle\!\rangle = \bigcap\left(\{\,[\![\phi]\!]\cup[\![\diamond\,\mathcal{D}']\!] \mid \phi\in\bigoplus\mathcal{D}\,\}\cup\langle\!\langle\mathcal{B}_n\rangle\!\rangle\right)\ .$$

This can also be written as $([\![\diamond\,\mathcal{D}']\!]\cup\bigcap\langle\!\langle\bigoplus\mathcal{D}\rangle\!\rangle)\cap\bigcap\langle\!\langle\mathcal{B}_n\rangle\!\rangle$. Substituting $[\![\diamond\,\mathcal{D}]\!]$ for $\bigcap\langle\!\langle\bigoplus\mathcal{D}\rangle\!\rangle$ and distributing $\cap$ over $\cup$ yields $([\![\diamond\,\mathcal{D}']\!]\cap[\![\mathcal{B}_n]\!])\cup([\![\diamond\,\mathcal{D}]\!]\cap[\![\mathcal{B}_n]\!])$. Finally, using (B1) and (B2.1) we can write this as

$$([\![\diamond\,\mathcal{D}\diamond\mathcal{B}_n]\!]\cap[\![\mathcal{B}_n]\!])\cup[\![(\diamond\,\mathcal{D})\cup\mathcal{B}_n]\!] = [\![\diamond\,\mathcal{D}\diamond\mathcal{B}_n]\!]\cup[\![(\diamond\,\mathcal{D})\cup\mathcal{B}_n]\!]$$
$$= [\![\diamond\,\mathcal{D}\diamond\mathcal{B}_n]\!] = [\![\diamond\,\mathcal{D}']\!]\ . \qquad\square$$

Appendix E.2. Formula-Based Update Operators

The set of possible remainders has a number of important properties from which properties of specific formula-based operators follow. We start with two auxiliary results which make it possible to construct a subset of a knowledge base with important properties on the semantic level.

Lemma 121. *Let $\mathcal{B}, \mathcal{U}$ be knowledge bases. Then $\mathcal{U}$ is consistent if and only if $\mathrm{rem}(\mathcal{B}, \mathcal{U})$ is non-empty.*

Proof. First suppose that $\mathcal{U}$ is consistent and let $\mathcal{R}$ be the set of all subsets $\mathcal{B}'$ of $\mathcal{B}$ such that $\mathcal{B}'\cup\mathcal{U}$ is consistent. $\mathcal{R}$ must be non-empty because $\emptyset$ clearly belongs to $\mathcal{R}$. So take some subset-maximal element $\mathcal{B}^*$ of $\mathcal{R}$. It is easy to see that $\mathcal{B}^*$ belongs to $\mathrm{rem}(\mathcal{B}, \mathcal{U})$.

On the other hand, if $\mathrm{rem}(\mathcal{B}, \mathcal{U})$ is non-empty, then it contains some set $\mathcal{B}'$ such that $\mathcal{B}'\cup\mathcal{U}$ is consistent. Thus it follows directly that $\mathcal{U}$ is also consistent. $\qquad\square$

Lemma 122. *Let $\mathcal{B}, \mathcal{C}$ be knowledge bases with $\langle\!\langle\mathcal{B}\rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle\mathcal{C}\rangle\!\rangle^{\mathfrak{I}}$, $\mathcal{B}'\subseteq\mathcal{B}$ and $\mathcal{C}' = \{\,\phi\in\mathcal{C} \mid [\![\phi]\!]\in\langle\!\langle\mathcal{B}'\rangle\!\rangle^{\mathfrak{I}}\,\}$. Then $\langle\!\langle\mathcal{B}'\rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle\mathcal{C}'\rangle\!\rangle^{\mathfrak{I}}$.*

Proof. Suppose first that $\mathcal{M}$ belongs to $\langle\!\langle\mathcal{B}'\rangle\!\rangle^{\mathfrak{I}}$. Then it also belongs to $\langle\!\langle\mathcal{B}\rangle\!\rangle^{\mathfrak{I}}$, so by our assumption either $\mathcal{M} = \mathfrak{I}$ or $\mathcal{M}$ belongs to $\langle\!\langle\mathcal{C}\rangle\!\rangle$. In the former case $\mathcal{M}$ belongs to $\langle\!\langle\mathcal{C}'\rangle\!\rangle^{\mathfrak{I}}$ and we are finished. In the latter case there is a formula $\phi \in \mathcal{C}$ such that $[\![\phi]\!] = \mathcal{M}$ and ϕ belongs to $\mathcal{C}'$ by its definition. Consequently, $\mathcal{M}$ belongs to $\langle\!\langle\mathcal{C}'\rangle\!\rangle^{\mathfrak{I}}$.

As for the other inclusion, if $\mathcal{M}$ belongs to $\langle\!\langle\mathcal{C}'\rangle\!\rangle^{\mathfrak{I}}$, then either $\mathcal{M} = \mathfrak{I}$ or for some formula $\phi \in \mathcal{C}'$ we have $\mathcal{M} = [\![\phi]\!]$. Therefore, $\mathcal{M}$ belongs to $\langle\!\langle\mathcal{B}'\rangle\!\rangle^{\mathfrak{I}}$ by the definition of $\mathcal{C}'$. $\qquad\square$

Lemma 123. *Let $\mathcal{B}, \mathcal{C}, \mathcal{U}, \mathcal{V}$ be knowledge bases such that $\langle\!\langle\mathcal{B}\rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle\mathcal{C}\rangle\!\rangle^{\mathfrak{I}}$ and $\langle\!\langle\mathcal{U}\rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle\mathcal{V}\rangle\!\rangle^{\mathfrak{I}}$, $\mathcal{B}' \subseteq \mathcal{B}$ such that $\mathcal{B}' \cup \mathcal{U}$ is consistent and $\mathcal{C}' = \{\, \phi \in \mathcal{C} \mid [\![\phi]\!] \in \langle\!\langle\mathcal{B}'\rangle\!\rangle^{\mathfrak{I}} \,\}$. Then $\mathcal{C}' \cup \mathcal{V}$ is consistent.*

Proof. To verify that $\mathcal{C}' \cup \mathcal{V}$ is consistent, we only need to use Lemma 122 and observe that

$$[\![\mathcal{C}' \cup \mathcal{V}]\!] = \bigcap\langle\!\langle\mathcal{C}' \cup \mathcal{V}\rangle\!\rangle = \bigcap(\langle\!\langle\mathcal{C}'\rangle\!\rangle \cup \langle\!\langle\mathcal{V}\rangle\!\rangle) = \bigcap(\langle\!\langle\mathcal{C}'\rangle\!\rangle^{\mathfrak{I}} \cup \langle\!\langle\mathcal{V}\rangle\!\rangle)$$

$$= \bigcap(\langle\!\langle\mathcal{B}'\rangle\!\rangle^{\mathfrak{I}} \cup \langle\!\langle\mathcal{U}\rangle\!\rangle) = \bigcap(\langle\!\langle\mathcal{B}'\rangle\!\rangle \cup \langle\!\langle\mathcal{U}\rangle\!\rangle) = \bigcap\langle\!\langle\mathcal{B}' \cup \mathcal{U}\rangle\!\rangle = [\![\mathcal{B}' \cup \mathcal{U}]\!] \;. \qquad\square$$

Proposition 124 (Syntax-Independence of Remainders). *Let $\mathcal{B}, \mathcal{C}, \mathcal{U}, \mathcal{V}$ be knowledge bases such that $\langle\!\langle\mathcal{B}\rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle\mathcal{C}\rangle\!\rangle^{\mathfrak{I}}$ and $\langle\!\langle\mathcal{U}\rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle\mathcal{V}\rangle\!\rangle^{\mathfrak{I}}$. Then $(\!(\mathrm{rem}(\mathcal{B},\mathcal{U}))\!)^{\mathfrak{I}} = (\!(\mathrm{rem}(\mathcal{C},\mathcal{V}))\!)^{\mathfrak{I}}$.*

Proof. We prove that $(\!(\mathrm{rem}(\mathcal{B},\mathcal{U}))\!)^{\mathfrak{I}} \subseteq (\!(\mathrm{rem}(\mathcal{C},\mathcal{V}))\!)^{\mathfrak{I}}$, the other inclusion follows by the same arguments since the formulation of the proposition is symmetric.

Take some $\mathcal{B}'$ from $\mathrm{rem}(\mathcal{B},\mathcal{U})$ and put $\mathcal{C}' = \{\, \phi \in \mathcal{C} \mid [\![\phi]\!] \in \langle\!\langle\mathcal{B}'\rangle\!\rangle^{\mathfrak{I}} \,\}$. We need to show that $\langle\!\langle\mathcal{B}'\rangle\!\rangle^{\mathfrak{I}}$ belongs to $(\!(\mathrm{rem}(\mathcal{C},\mathcal{V}))\!)^{\mathfrak{I}}$. Due to Lemma 122, $\langle\!\langle\mathcal{B}'\rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle\mathcal{C}'\rangle\!\rangle^{\mathfrak{I}}$, so it suffices to prove that $\mathcal{C}'$ belongs to $\mathrm{rem}(\mathcal{C},\mathcal{V})$. First, note that $\mathcal{C}'$ is clearly a subset of $\mathcal{C}$ and, by Lemma 123, $\mathcal{C}' \cup \mathcal{V}$ is consistent. We prove that $\mathcal{C}'$ is subset-maximal with these properties by contradiction. Suppose that $\mathcal{C}^*$ is such that $\mathcal{C}' \subsetneq \mathcal{C}^* \subseteq \mathcal{C}$ and $\mathcal{C}^* \cup \mathcal{V}$ is consistent and let $\mathcal{B}^* = \{\, \phi \in \mathcal{B} \mid [\![\phi]\!] \in \langle\!\langle\mathcal{C}^*\rangle\!\rangle^{\mathfrak{I}} \,\}$. Clearly, $\mathcal{B}^*$ is a subset of $\mathcal{B}$ and, by Lemma 123, $\mathcal{B}^* \cup \mathcal{U}$ is consistent. To reach a conflict, we need to show that $\mathcal{B}'$ is a proper subset of $\mathcal{B}^*$. First note that $\langle\!\langle\mathcal{C}'\rangle\!\rangle^{\mathfrak{I}}$ cannot be equal to $\langle\!\langle\mathcal{C}^*\rangle\!\rangle^{\mathfrak{I}}$ – if it were, then for every formula $\phi \in \mathcal{C}^*$ it would hold that ϕ belongs to $\mathcal{C}$ and $[\![\phi]\!]$ belongs to $\langle\!\langle\mathcal{B}'\rangle\!\rangle^{\mathfrak{I}}$, so ϕ belongs to $\mathcal{C}'$ by its definition, contrary to the assumption that $\mathcal{C}'$ is a proper subset of $\mathcal{C}^*$. This, together with Lemma 122, implies that

$$\langle\!\langle\mathcal{B}'\rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle\mathcal{C}'\rangle\!\rangle^{\mathfrak{I}} \subsetneq \langle\!\langle\mathcal{C}^*\rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle\mathcal{B}^*\rangle\!\rangle^{\mathfrak{I}} \;. \tag{E.2}$$

It immediately follows that $\mathcal{B}' \neq \mathcal{B}^*$. Furthermore, for any formula ϕ from $\mathcal{B}'$, ϕ belongs to $\mathcal{B}$ and it follows from (E.2) that $[\![\phi]\!]$ belongs to $\langle\!\langle\mathcal{C}^*\rangle\!\rangle^{\mathfrak{I}}$, so ϕ belongs to $\mathcal{B}^*$ by its definition. This means that $\mathcal{B}'$ is a proper subset of $\mathcal{B}^*$, contrary to the assumption that $\mathcal{B}'$ belongs to $\mathrm{rem}(\mathcal{B},\mathcal{U})$. $\qquad\square$

Lemma 125 (Equivalent Formulas in Remainders). *Let $\mathcal{B}, \mathcal{U}$ be knowledge bases, $\phi, \psi \in \mathcal{B}$ formulas such that $[\![\phi]\!] = [\![\psi]\!]$ and $\mathcal{B}' \in \mathrm{rem}(\mathcal{B},\mathcal{U})$ a possible remainder. Then $\phi \in \mathcal{B}'$ if and only if $\psi \in \mathcal{B}'$.*

Proof. Without loss of generality, assume that ϕ belongs to $\mathcal{B}'$ but ψ does not. Then $\mathcal{B}' \cup \{\, \psi \,\}$ is a subset of $\mathcal{B}$ that is consistent with $\mathcal{U}$. This is in conflict with the maximality of $\mathcal{B}'$. $\qquad\square$

Corollary 126. *Let $\mathcal{B}, \mathcal{U}$ be knowledge bases and $\mathcal{R} \subseteq \mathrm{rem}(\mathcal{B},\mathcal{U})$ a set of possible remainders. Then $\bigcap(\!(\mathcal{R})\!) = \langle\!\langle\bigcap\mathcal{R}\rangle\!\rangle$.*

Proof. First suppose that $\mathcal{M}$ belongs to $\bigcap(\!(\mathcal{R})\!)$ and take some $\mathcal{B}' \in \mathcal{R}$ and some formula $\phi \in \mathcal{B}'$ such that $[\![\phi]\!] = \mathcal{M}$. Now take an arbitrary $\mathcal{B}^* \in \mathcal{R}$. Since $\mathcal{M}$ belongs to $\langle\!\langle\mathcal{B}^*\rangle\!\rangle$, there must exist a formula $\psi \in \mathcal{B}^*$ such that $[\![\psi]\!] = \mathcal{M}$. Consequently, $[\![\phi]\!] = [\![\psi]\!]$ and by Lemma 125 we obtain that ϕ also belongs to $\mathcal{B}^*$. Thus, ϕ belongs to $\bigcap\mathcal{R}$ and $\mathcal{M}$ belongs to $\langle\!\langle\bigcap\mathcal{R}\rangle\!\rangle$.

On the other hand, if $\mathcal{M}$ belongs to $\langle\!\langle\bigcap\mathcal{R}\rangle\!\rangle$, then there is a formula $\phi \in \bigcap\mathcal{R}$ such that $[\![\phi]\!] = \mathcal{M}$. Consequently, $\mathcal{M}$ belongs to all members of $(\!(\mathcal{R})\!)$, thus also belongs to their intersection. $\qquad\square$

Corollary 127. *Let $\mathcal{B}$, $\mathcal{U}$ be knowledge bases, $\mathcal{R} \subseteq \mathrm{rem}(\mathcal{B},\mathcal{U})$ a set of possible remainders. Then, $\bigcap(\langle\!\langle\mathcal{R}\rangle\!\rangle)^{\mathfrak{I}} = \langle\!\langle\bigcap\mathcal{R}\rangle\!\rangle^{\mathfrak{I}}$.*

Proof. Follows from Corollary 126 and from the fact that $\mathfrak{I}$ belongs to both sides of the equation. $\qquad\square$

Proposition 128 (Properties of the WIDTIO Operator). *The WIDTIO operator satisfies (F1), (F2.1) and (F4).*

Proof. By definition $\mathcal{U} \subseteq \mathcal{B} \circ_{\mathsf{widtio}} \mathcal{U}$ and (F1) is obtained by applying $\langle\!\langle\cdot\rangle\!\rangle$ to both sides of this inclusion.

In order to verify that (F2.1) holds, suppose that $\mathcal{M}$ belongs to $\langle\!\langle\mathcal{B} \circ_{\mathsf{widtio}} \mathcal{U}\rangle\!\rangle$. Then there is some formula ϕ from $\mathcal{U} \cup \bigcap\mathrm{rem}(\mathcal{B},\mathcal{U})$ such that $[\![\phi]\!] = \mathcal{M}$. If ϕ belongs to $\mathcal{U}$, then it immediately follows that $\mathcal{M}$ belongs to $\langle\!\langle\mathcal{U}\rangle\!\rangle$, and consequently also to $\langle\!\langle\mathcal{B} \cup \mathcal{U}\rangle\!\rangle$. If ϕ belongs to $\mathcal{B}'$ for all $\mathcal{B}' \in \mathrm{rem}(\mathcal{B},\mathcal{U})$, then ϕ also belongs to $\mathcal{B}$. Thus, $\mathcal{M}$ is a member of $\langle\!\langle\mathcal{B}\rangle\!\rangle$, and consequently also of $\langle\!\langle\mathcal{B} \cup \mathcal{U}\rangle\!\rangle$.

Finally, to verify (F4), suppose that $\langle\!\langle\mathcal{B}\rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle\mathcal{C}\rangle\!\rangle^{\mathfrak{I}}$ and $\langle\!\langle\mathcal{U}\rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle\mathcal{V}\rangle\!\rangle^{\mathfrak{I}}$. The following follows from the definition of the WIDTIO operator, Corollary 127 and Proposition 124.

$$\langle\!\langle\mathcal{B} \circ_{\mathsf{widtio}} \mathcal{U}\rangle\!\rangle^{\mathfrak{I}} = \left\langle\!\!\left\langle\mathcal{U} \cup \bigcap\mathrm{rem}(\mathcal{B},\mathcal{U})\right\rangle\!\!\right\rangle^{\mathfrak{I}} = \langle\!\langle\mathcal{U}\rangle\!\rangle^{\mathfrak{I}} \cup \left\langle\!\!\left\langle\bigcap\mathrm{rem}(\mathcal{B},\mathcal{U})\right\rangle\!\!\right\rangle^{\mathfrak{I}}$$

$$= \langle\!\langle\mathcal{U}\rangle\!\rangle^{\mathfrak{I}} \cup \bigcap(\langle\!\langle\mathrm{rem}(\mathcal{B},\mathcal{U})\rangle\!\rangle)^{\mathfrak{I}} = \langle\!\langle\mathcal{V}\rangle\!\rangle^{\mathfrak{I}} \cup \bigcap(\langle\!\langle\mathrm{rem}(\mathcal{C},\mathcal{V})\rangle\!\rangle)^{\mathfrak{I}}$$

$$= \langle\!\langle\mathcal{V}\rangle\!\rangle^{\mathfrak{I}} \cup \left\langle\!\!\left\langle\bigcap\mathrm{rem}(\mathcal{C},\mathcal{V})\right\rangle\!\!\right\rangle^{\mathfrak{I}} = \left\langle\!\!\left\langle\mathcal{V} \cup \bigcap\mathrm{rem}(\mathcal{C},\mathcal{V})\right\rangle\!\!\right\rangle^{\mathfrak{I}} = \langle\!\langle\mathcal{C} \circ_{\mathsf{widtio}} \mathcal{V}\rangle\!\rangle^{\mathfrak{I}} \ . \qquad\square$$

Proposition 129 (Properties of Regular Bold Operators). *Regular Bold operators satisfy (F1), (F2.1) and (F4).*

Proof. By definition $\mathcal{U} \subseteq \mathcal{B} \circ_{\mathsf{bold}}^{s} \mathcal{U}$ and (F1) is obtained by applying $\langle\!\langle\cdot\rangle\!\rangle$ to both sides of this inclusion.

In order to verify that (F2.1) holds, suppose that $\mathcal{M}$ belongs to $\langle\!\langle\mathcal{B} \circ_{\mathsf{bold}}^{s} \mathcal{U}\rangle\!\rangle$. Then there is some formula ϕ from $\mathcal{U} \cup s(\mathrm{rem}(\mathcal{B},\mathcal{U}))$ such that $[\![\phi]\!] = \mathcal{M}$. If ϕ belongs to $\mathcal{U}$, then it immediately follows that $\mathcal{M}$ belongs to $\langle\!\langle\mathcal{U}\rangle\!\rangle$, and consequently also to $\langle\!\langle\mathcal{B} \cup \mathcal{U}\rangle\!\rangle$. If ϕ belongs to $\mathcal{B}' = s(\mathrm{rem}(\mathcal{B},\mathcal{U}))$, then ϕ also belongs to $\mathcal{B}$. Thus, $\mathcal{M}$ is a member of $\langle\!\langle\mathcal{B}\rangle\!\rangle$, and consequently also of $\langle\!\langle\mathcal{B} \cup \mathcal{U}\rangle\!\rangle$.

Finally, to verify (F4), suppose that $\langle\!\langle\mathcal{B}\rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle\mathcal{C}\rangle\!\rangle^{\mathfrak{I}}$ and $\langle\!\langle\mathcal{U}\rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle\mathcal{V}\rangle\!\rangle^{\mathfrak{I}}$. The following follows from the definition of the WIDTIO operator, Proposition 124 and the regularity property of $\circ_{\mathsf{bold}}^{s}$.

$$\langle\!\langle\mathcal{B} \circ_{\mathsf{bold}}^{s} \mathcal{U}\rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle\mathcal{U} \cup s(\mathrm{rem}(\mathcal{B},\mathcal{U}))\rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle\mathcal{U}\rangle\!\rangle^{\mathfrak{I}} \cup \langle\!\langle s(\mathrm{rem}(\mathcal{B},\mathcal{U}))\rangle\!\rangle^{\mathfrak{I}}$$

$$= \langle\!\langle\mathcal{V}\rangle\!\rangle^{\mathfrak{I}} \cup \langle\!\langle s(\mathrm{rem}(\mathcal{C},\mathcal{V}))\rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle\mathcal{V} \cup s(\mathrm{rem}(\mathcal{C},\mathcal{V}))\rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle\mathcal{C} \circ_{\mathsf{bold}}^{s} \mathcal{V}\rangle\!\rangle^{\mathfrak{I}} \ . \qquad\square$$

Proposition 130 (Properties of the Cross-Product Operator). *The Cross-Product operator satisfies (F1), (B2.1) and (F4) but does not satisfy (F2.1).*

Proof. By definition $\mathcal{U} \subseteq \mathcal{B} \circ_{\mathsf{cp}} \mathcal{U}$ and (F1) is obtained by applying $\langle\!\langle\cdot\rangle\!\rangle$ to both sides of this inclusion.

To see that $\circ_{\mathsf{cp}}$ does not satisfy (F2.1), note that $\{\,p,q\,\} \circ_{\mathsf{cp}} \{\,\neg p \vee \neg q\,\} = \{\,p \vee q, \neg p \vee \neg q\,\}$ and $[\![p \vee q]\!]$ does not belong to $\langle\!\langle\{\,p,q,\neg p \vee \neg q\,\}\rangle\!\rangle$.

In order to verify (B2.1), take some I from $[\![\mathcal{B} \cup \mathcal{U}]\!]$. We need to show that I is a model of $\mathcal{B} \circ_{\mathsf{cp}} \mathcal{U}$. Obviously, I is a model of $\mathcal{U}$, so it remains to prove that I is a model of the formula

$$\psi = \bigvee_{\mathcal{B}' \in \mathrm{rem}(\mathcal{B},\mathcal{U})} \ \bigwedge_{\phi \in \mathcal{B}'} \phi \ .$$

Since I is a model of $\mathcal{U}$, we conclude that $\mathcal{U}$ is consistent, so according to Lemma 121, $\mathrm{rem}(\mathcal{B}, \mathcal{U})$ is non-empty. Take some $\mathcal{B}^*$ from $\mathrm{rem}(\mathcal{B}, \mathcal{U})$. We obtain the following: $\llbracket \psi \rrbracket = \bigcup_{\mathcal{B}' \in \mathrm{rem}(\mathcal{B}, \mathcal{U})} \llbracket \mathcal{B}' \rrbracket \supseteq \llbracket \mathcal{B}^* \rrbracket \supseteq \llbracket \mathcal{B} \rrbracket$. Hence, since I belongs to $\llbracket \mathcal{B} \rrbracket$, it also belongs to $\llbracket \psi \rrbracket$.

Finally, to verify (F4), suppose that $\langle\!\langle \mathcal{B} \rangle\!\rangle^{\mathcal{I}} = \langle\!\langle \mathcal{C} \rangle\!\rangle^{\mathcal{I}}$ and $\langle\!\langle \mathcal{U} \rangle\!\rangle^{\mathcal{I}} = \langle\!\langle \mathcal{V} \rangle\!\rangle^{\mathcal{I}}$ and take some $\mathcal{M}$ from $\langle\!\langle \mathcal{B} \circ_{\mathsf{cp}} \mathcal{U} \rangle\!\rangle^{\mathcal{I}}$. In the trivial case when $\mathcal{M} = \mathcal{I}$ it immediately follows that $\mathcal{M}$ belongs to $\langle\!\langle \mathcal{C} \circ_{\mathsf{cp}} \mathcal{V} \rangle\!\rangle^{\mathcal{I}}$. Otherwise, there is a formula ϕ from $\mathcal{U} \cup \{ \psi \}$ such that $\llbracket \phi \rrbracket = \mathcal{M}$. If ϕ belongs to $\mathcal{U}$, then $\mathcal{M}$ belongs to $\langle\!\langle \mathcal{U} \rangle\!\rangle$ and by assumption also to $\langle\!\langle \mathcal{V} \rangle\!\rangle^{\mathcal{I}}$. By (F1) we then obtain that $\mathcal{M}$ belongs to $\langle\!\langle \mathcal{C} \circ_{\mathsf{cp}} \mathcal{V} \rangle\!\rangle^{\mathcal{I}}$. On the other hand, if ϕ is ψ, then due to Proposition 124, $(\!(\mathrm{rem}(\mathcal{B}, \mathcal{U}))\!)^{\mathcal{I}} = (\!(\mathrm{rem}(\mathcal{C}, \mathcal{V}))\!)^{\mathcal{I}}$, so

$$\llbracket \phi \rrbracket = \llbracket \psi \rrbracket = \bigcup_{\mathcal{B}' \in \mathrm{rem}(\mathcal{B}, \mathcal{U})} \llbracket \mathcal{B}' \rrbracket = \bigcup_{\mathcal{C}' \in \mathrm{rem}(\mathcal{C}, \mathcal{V})} \llbracket \mathcal{C}' \rrbracket = \llbracket \psi' \rrbracket$$

where $\mathcal{C} \circ_{\mathsf{cp}} \mathcal{V} = \mathcal{V} \cup \{ \psi' \}$. Therefore, $\llbracket \phi \rrbracket$ belongs to $\langle\!\langle \mathcal{C} \circ_{\mathsf{cp}} \mathcal{V} \rangle\!\rangle^{\mathcal{I}}$. The proof of the other inclusion is symmetric. $\qquad \square$

Proof of Proposition 36. Follows from Propositions 128, 129 and 130. $\qquad \square$

Proposition 131. *If $\circ$ is an update operator that satisfies (F1), (F2.1) and (F4), then there exists an exception function ε such that for every ε-based update operator $\oplus$ and all finite sequences of knowledge bases $\mathcal{D}$,* $\langle\!\langle \bigcirc \mathcal{D} \rangle\!\rangle^{\mathcal{I}} = \langle\!\langle \bigoplus \mathcal{D} \rangle\!\rangle^{\mathcal{I}}$.

Proof. Let the exception function ε be defined for all sets of interpretations $\mathcal{M}$ and all sets of sets of interpretations $\mathcal{S}, \mathcal{T}$ as

$$\varepsilon(\mathcal{M}, \mathcal{S}, \mathcal{T}) = \begin{cases} \emptyset & \mathcal{M} \in \langle\!\langle \mathcal{B} \circ \mathcal{U} \rangle\!\rangle^{\mathcal{I}} \ ; \\ \mathcal{I} & \mathcal{M} \notin \langle\!\langle \mathcal{B} \circ \mathcal{U} \rangle\!\rangle^{\mathcal{I}} \ , \end{cases}$$

where $\mathcal{B}, \mathcal{U}$ are some knowledge bases such that $\langle\!\langle \mathcal{B} \rangle\!\rangle^{\mathcal{I}} = \mathcal{S} \cup \{ \mathcal{I} \}$ and $\langle\!\langle \mathcal{U} \rangle\!\rangle^{\mathcal{I}} = \mathcal{T} \cup \{ \mathcal{I} \}$. Note that this definition is unambiguous since the existence of such $\mathcal{B}$ and $\mathcal{U}$ is guaranteed and regardless of which pair of knowledge bases with these properties we choose, we obtain the same result due to the assumption that $\circ$ satisfies (F4). Take some ε-based operator $\oplus$. We proceed by induction on the length n of $\mathcal{D} = \langle \mathcal{B}_i \rangle_{i < n}$.

1° If $n = 0$, then it immediately follows that $\langle\!\langle \diamond \mathcal{D} \rangle\!\rangle = \langle\!\langle \diamond \langle \mathcal{B}_0 \rangle \rangle\!\rangle = \langle\!\langle \mathcal{B}_0 \rangle\!\rangle = \langle\!\langle \bigoplus \langle \mathcal{B}_0 \rangle \rangle\!\rangle = \langle\!\langle \bigoplus \mathcal{D} \rangle\!\rangle$.

2° Suppose that the claim holds for n, i.e. for $\mathcal{D} = \langle \mathcal{B}_i \rangle_{i < n}$ we have $\langle\!\langle \diamond \mathcal{D} \rangle\!\rangle^{\mathcal{I}} = \langle\!\langle \bigoplus \mathcal{D} \rangle\!\rangle^{\mathcal{I}}$. Our goal is to show that it also holds for $n + 1$, i.e. for $\mathcal{D}' = \langle \mathcal{B}_i \rangle_{i < n+1}$. It follows that

$$\langle\!\langle \bigoplus \mathcal{D}' \rangle\!\rangle = \langle\!\langle \bigoplus \mathcal{D} \oplus \mathcal{B}_n \rangle\!\rangle = \{\, \llbracket \phi \rrbracket \cup \varepsilon(\llbracket \phi \rrbracket, \langle\!\langle \bigoplus \mathcal{D} \rangle\!\rangle, \langle\!\langle \mathcal{B}_n \rangle\!\rangle) \mid \phi \in \bigoplus \mathcal{D} \,\} \cup \langle\!\langle \mathcal{B}_n \rangle\!\rangle \ .$$

Thus, $\langle\!\langle \bigoplus \mathcal{D}' \rangle\!\rangle^{\mathcal{I}} = \{\, \llbracket \phi \rrbracket \cup \varepsilon(\llbracket \phi \rrbracket, \langle\!\langle \bigoplus \mathcal{D} \rangle\!\rangle, \langle\!\langle \mathcal{B}_n \rangle\!\rangle) \mid \phi \in \bigoplus \mathcal{D} \,\} \cup \langle\!\langle \mathcal{B}_n \rangle\!\rangle \cup \{ \mathcal{I} \}$ which in turn can be written as

$$\left\{\, \mathcal{M} \ \middle| \ \mathcal{M} \in \langle\!\langle \bigoplus \mathcal{D} \rangle\!\rangle \cap \langle\!\langle \bigoplus \mathcal{D} \circ \mathcal{B}_n \rangle\!\rangle^{\mathcal{I}} \,\right\} \cup \left\{\, \mathcal{I} \ \middle| \ \mathcal{M} \in \langle\!\langle \bigoplus \mathcal{D} \rangle\!\rangle \setminus \langle\!\langle \bigoplus \mathcal{D} \circ \mathcal{B}_n \rangle\!\rangle^{\mathcal{I}} \,\right\} \cup \langle\!\langle \mathcal{B}_n \rangle\!\rangle \cup \{ \mathcal{I} \}$$

and simplified into $\left(\langle\!\langle \bigoplus \mathcal{D} \rangle\!\rangle^{\mathcal{I}} \cap \langle\!\langle \bigoplus \mathcal{D} \circ \mathcal{B}_n \rangle\!\rangle^{\mathcal{I}} \right) \cup \langle\!\langle \mathcal{B}_n \rangle\!\rangle^{\mathcal{I}}$. Since $\circ$ satisfies (F4), it follows from the inductive assumption that $\langle\!\langle \bigoplus \mathcal{D} \circ \mathcal{B}_n \rangle\!\rangle^{\mathcal{I}} = \langle\!\langle \bigcirc \mathcal{D} \circ \mathcal{B}_n \rangle\!\rangle^{\mathcal{I}}$. Thus, we obtain the set

$$\left(\langle\!\langle \bigcirc \mathcal{D} \rangle\!\rangle^{\mathcal{I}} \cap \langle\!\langle \bigcirc \mathcal{D} \circ \mathcal{B}_n \rangle\!\rangle^{\mathcal{I}} \right) \cup \langle\!\langle \mathcal{B}_n \rangle\!\rangle^{\mathcal{I}}$$

and by distributing $\cup$ over $\cap$ and using (F1) and (F2.1) we obtain

$$\left(\langle\!\langle \bigcirc \mathcal{D} \rangle\!\rangle^{\mathcal{I}} \cup \langle\!\langle \mathcal{B}_n \rangle\!\rangle^{\mathcal{I}} \right) \cap \left(\langle\!\langle \bigcirc \mathcal{D} \circ \mathcal{B}_n \rangle\!\rangle^{\mathcal{I}} \cup \langle\!\langle \mathcal{B}_n \rangle\!\rangle^{\mathcal{I}} \right) = \langle\!\langle \bigcirc \mathcal{D} \cup \mathcal{B}_n \rangle\!\rangle^{\mathcal{I}} \cap \langle\!\langle \bigcirc \mathcal{D} \circ \mathcal{B}_n \rangle\!\rangle^{\mathcal{I}}$$

$$= \langle\!\langle \bigcirc \mathcal{D} \circ \mathcal{B}_n \rangle\!\rangle^{\mathcal{I}} = \langle\!\langle \bigcirc \mathcal{D}' \rangle\!\rangle^{\mathcal{I}} \ . \qquad \square$$

Proposition 132. *If ∘ an update operator that satisfies (F1), (B2.1) and (F4), then there exists an exception function ε such that for every ε-based update operator $\oplus$ and all knowledge bases $\mathcal{B}, \mathcal{U}$, $[\![\mathcal{B} \circ \mathcal{U}]\!] = [\![\mathcal{B} \oplus \mathcal{U}]\!]$.*

Proof. Let the exception function ε be defined for all sets of interpretations $\mathcal{M}$ and all sets of sets of interpretations $\mathcal{S}, \mathcal{T}$ as

$$\varepsilon(\mathcal{M}, \mathcal{S}, \mathcal{T}) = [\![\mathcal{B} \circ \mathcal{U}]\!] \ ,$$

where $\mathcal{B}, \mathcal{U}$ are some knowledge bases such that $\langle\!\langle\mathcal{B}\rangle\!\rangle^{\mathfrak{I}} = \mathcal{S} \cup \{\mathfrak{I}\}$ and $\langle\!\langle\mathcal{U}\rangle\!\rangle^{\mathfrak{I}} = \mathcal{T} \cup \{\mathfrak{I}\}$. Note that this definition is unambiguous since the existence of such $\mathcal{B}$ and $\mathcal{U}$ is guaranteed and regardless of which pair of knowledge bases with these properties we choose, we obtain the same result due to the assumption that ∘ satisfies (F4).

Take some ε-based operator $\oplus$. Then $[\![\mathcal{B} \oplus \mathcal{U}]\!] = \bigcap (\{\, [\![\phi]\!] \cup \varepsilon([\![\phi]\!], \langle\!\langle\mathcal{B}\rangle\!\rangle, \langle\!\langle\mathcal{U}\rangle\!\rangle) \mid \phi \in \mathcal{B} \,\} \cup \langle\!\langle\mathcal{U}\rangle\!\rangle)$, which can be written as $\bigcap \{\, [\![\phi]\!] \cup [\![\mathcal{B} \circ \mathcal{U}]\!] \mid \phi \in \mathcal{B} \,\} \cap \bigcap \langle\!\langle\mathcal{U}\rangle\!\rangle$ and simplified into $([\![\mathcal{B} \circ \mathcal{U}]\!] \cup [\![\mathcal{B}]\!]) \cap [\![\mathcal{U}]\!]$. Furthermore, due to (B1) and (B2.1),

$$([\![\mathcal{B} \circ \mathcal{U}]\!] \cup [\![\mathcal{B}]\!]) \cap [\![\mathcal{U}]\!] = ([\![\mathcal{B} \circ \mathcal{U}]\!] \cap [\![\mathcal{U}]\!]) \cup ([\![\mathcal{B}]\!] \cap [\![\mathcal{U}]\!]) = [\![\mathcal{B} \circ \mathcal{U}]\!] \cup [\![\mathcal{B} \cup \mathcal{U}]\!] = [\![\mathcal{B} \circ \mathcal{U}]\!] \ . \quad \square$$

Proof of Theorem 37. Follows from Propositions 131 and 132. $\hfill\square$

References

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

[2] J. J. Alferes and L. M. Pereira. Update-programs can update programs. In J. Dix, L. M. Pereira, and T. C. Przymusinski, editors, *Non-Monotonic Extensions of Logic Programming (NMELP '96), Selected Papers*, volume 1216 of *Lecture Notes in Computer Science*, pages 110–131, Bad Honnef, Germany, September 5-6 1996. Springer. ISBN 3-540-62843-6.

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

[4] J. J. Alferes, A. Brogi, J. A. Leite, and L. M. Pereira. An evolvable rule-based e-mail agent. In F. Moura-Pires and S. Abreu, editors, *Proceedings of the 11th Portuguese Conference Artificial Intelligence (EPIA 2003)*, volume 2902 of *Lecture Notes in Computer Science*, pages 394–408, Beja, Portugal, December 4-7 2003. Springer. ISBN 3-540-20589-6.

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

[6] K. R. Apt and M. Bezem. Acyclic programs. *New Generation Computing*, 9(3/4):335–364, 1991.

[7] K. R. Apt, H. A. Blair, and A. Walker. Towards a theory of declarative knowledge. In J. Minker, editor, *Foundations of Deductive Databases and Logic Programming*, pages 89–148. Morgan Kaufmann, San Francisco, CA, USA, 1988. ISBN 0-934613-40-0.

[8] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2nd edition, 2007.

[9] Y. Bong. The description logic ABox update problem revisited. Master's thesis, Dresden University of Technology, Dresden, Germany, February 2007.

[10] C. Boutilier. Generalized update: Belief change in dynamic settings. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI 95)*, pages 1550–1556, Montral, Québec, Canada, August 20-25 1995. Morgan Kaufmann.

[11] G. Brewka and J. Hertzberg. How to do things with worlds: On formalizing actions and plans. *Journal of Logic and Computation*, 3(5):517–532, 1993.

[12] P. Cabalar, D. Pearce, and A. Valverde. Minimal logic programs. In V. Dahl and I. Niemelä, editors, *Proceedings of the 23rd International Conference on Logic Programming (ICLP 2007)*, volume 4670 of *Lecture Notes in Computer Science*, pages 104–118, Porto, Portugal, September 8-13 2007. Springer. ISBN 978-3-540-74608-9.

[13] D. Calvanese, E. Kharlamov, W. Nutt, and D. Zheleznyakov. Evolution of DL-Lite knowledge bases. In P. F. Patel-Schneider, Y. Pan, P. Hitzler, P. Mika, L. Zhang, J. Z. Pan, I. Horrocks, and B. Glimm, editors, *International Semantic Web Conference (1)*, volume 6496 of *Lecture Notes in Computer Science*, pages 112–128, Shanghai, China, November 7-11 2010. Springer. ISBN 978-3-642-17745-3.

[14] A. Colmerauer, H. Kanoui, P. Roussel, and R. Pasero. Un systéme de communication homme-machine en français. Technical report, Groupe de Recherche en Intelligence Artificielle, Université d'Aix-Marseille II, 1973.

[15] J. de Bruijn, D. Pearce, A. Polleres, and A. Valverde. A semantical framework for hybrid knowledge bases. *Journal of Knowledge and Information Systems*, 25(1):81–104, 2010.

[16] J. de Bruijn, T. Eiter, A. Polleres, and H. Tompits. Embedding nonground logic programs into autoepistemic logic for knowledge-base combination. *ACM Transactions on Computational Logic (TOCL)*, 12(3):20, 2011.

[17] G. De Giacomo, M. Lenzerini, A. Poggi, and R. Rosati. On the update of description logic ontologies at the instance level. In *Proceedings of the 21st National Conference on Artificial Intelligence and the 18th Innovative Applications of Artificial Intelligence Conference*, Boston, Massachusetts, USA, July 16-20 2006. AAAI Press.

[18] G. De Giacomo, M. Lenzerini, A. Poggi, and R. Rosati. On the approximation of instance level update and erasure in description logics. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI 2007)*, pages 403–408, Vancouver, British Columbia, Canada, July 22-26 2007. AAAI Press. ISBN 978-1-57735-323-2.

[19] G. De Giacomo, M. Lenzerini, A. Poggi, and R. Rosati. On instance-level update and erasure in description logic ontologies. *Journal of Logic and Computation*, 19(5):745–770, 2009.

[20] J. P. Delgrande, T. Schaub, and H. Tompits. A preference-based framework for updating logic programs. In C. Baral, G. Brewka, and J. S. Schlipf, editors, *Proceedings of the 9th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2007)*, volume 4483 of *Lecture Notes in Computer Science*, pages 71–83, Tempe, AZ, USA, May 15-17 2007. Springer. ISBN 978-3-540-72199-4.

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

[22] P. Doherty, W. Lukaszewicz, and E. Madalinska-Bugaj. The PMA and relativizing minimal change for action update. In A. G. Cohn, L. K. Schubert, and S. C. Shapiro, editors, *Proceedings of the 6th International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 258–269, Trento, Italy, June 2-5 1998. Morgan Kaufmann.

[23] C. Drescher, H. Liu, F. Baader, P. Steinke, and M. Thielscher. Putting ABox updates into action. In *Proceedings of the 8th IJCAI International Workshop on Nonmontonic Reasoning, Action and Change (NRAC-09)*, 2009.

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

[25] R. Fagin, J. D. Ullman, and M. Y. Vardi. On the semantics of updates in databases. In *Proceedings of the 2nd ACM SIGACT-SIGMOD Symposium on Principles of Database Systems (PODS 1986)*, pages 352–365, Colony Square Hotel, Atlanta, Georgia, March 21-23 1983. ACM. ISBN 0-89791-097-4.

[26] G. Flouris, D. Makanatas, H. Kondylakis, D. Plexousakis, and G. Antoniou. Ontology change: classification and survey. *The Knowledge Engineering Review*, 23(2):117–152, 2008.

[27] P. Gärdenfors. *Belief Revision*, chapter Belief Revision: An Introduction, pages 1–28. Cambridge University Press, 1992.

[28] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In R. A. Kowalski and K. A. Bowen, editors, *Proceedings of the 5th International Conference and Symposium on Logic Programming (ICLP/SLP 1988)*, pages 1070–1080, Seattle, Washington, August 15-19 1988. MIT Press.

[29] M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9 (3-4):365–385, 1991.

[30] M. L. Ginsberg. Counterfactuals. *Artificial Intelligence*, 30(1):35–79, 1986.

[31] M. L. Ginsberg and D. E. Smith. Reasoning about action i: A possible worlds approach. *Artificial Intelligence*, 35(2): 165–195, 1988.

[32] C. Halaschek-Wiener and Y. Katz. Belief base revision for expressive description logics. In B. C. Grau, P. Hitzler, C. Shankey, and E. Wallace, editors, *Proceedings of the OWLED*06 Workshop on OWL: Experiences and Directions*, volume 216 of *CEUR Workshop Proceedings*, Athens, Georgia, USA, November 10-11 2006. CEUR-WS.org.

[33] S. O. Hansson. Reversing the Levi identity. *Journal of Philosophical Logic*, 22(6):637–669, December 1993.

[34] A. Herzig and O. Rifi. Propositional belief base update and minimal change. *Artificial Intelligence*, 115(1):107–138, 1999.

[35] P. Hitzler and B. Parsia. Ontologies and rules. In S. Staab and R. Studer, editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 111–132. Springer, Berlin, second edition, 2009. ISBN 978-3-540-70999-2.

[36] M. Homola. Dynamic logic programming: Various semantics are equal on acyclic programs. In J. A. Leite and P. Torroni, editors, *Proceedings of the 5th International Workshop on Computational Logic in Multi-Agent Systems (CLIMA V)*, volume 3487 of *Lecture Notes in Computer Science*, pages 78–95, Lisbon, Portugal, September 29-30 2004. Springer. ISBN 3-540-28060-X.

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

[38] K. Inoue and C. Sakama. Negation as failure in the head. *Journal of Logic Programming*, 35(1):39–78, 1998.

[39] K. Inoue and C. Sakama. Equivalence of logic programs under updates. In J. J. Alferes and J. A. Leite, editors, *Proceedings of the 9th European Conference on Logics in Artificial Intelligence (JELIA 2004)*, volume 3229 of *Lecture Notes in Computer Science*, pages 174–186, Lisbon, Portugal, September 27-30 2004. Springer. ISBN 3-540-23242-7.

[40] T. Janhunen. On the effect of default negation on the expressiveness of disjunctive rules. In T. Eiter, W. Faber, and M. Truszczynski, editors, *Proceedings of the 6th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2001)*, volume 2173 of *Lecture Notes in Computer Science*, pages 93–106, Vienna, Austria, September 17-19 2001. Springer. ISBN 3-540-42593-4.

[41] H. Katsuno and A. O. Mendelzon. On the difference between updating a knowledge base and revising it. In J. F. Allen, R. Fikes, and E. Sandewall, editors, *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, pages 387–394, Cambridge, MA, USA, April 22-25 1991. Morgan Kaufmann Publishers. ISBN 1-55860-165-1.

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

[43] E. Kharlamov, D. Zheleznyakov, and D. Calvanese. Capturing model-based ontology evolution at the instance level: The case of dl-lite. *Journal of Computer and System Sciences*, 79(6):835–872, 2013.

[44] M. Knorr, J. J. Alferes, and P. Hitzler. Local closed world reasoning with description logics under the well-founded semantics. *Artificial Intelligence*, 175(9-10):1528–1554, 2011.

[45] R. A. Kowalski. Predicate logic as programming language. In *IFIP Congress*, pages 569–574, 1974.

[46] P. Krümpelmann. Dependency semantics for sequences of extended logic programs. *Logic Journal of the IGPL*, 20(5): 943–966, 2012.

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

[48] J. A. Leite and L. M. Pereira. Generalizing updates: From models to programs. In J. Dix, L. M. Pereira, and T. C. Przymusinski, editors, *Proceedings of the 3rd International Workshop on Logic Programming and Knowledge Representation (LPKR '97), October 17, 1997, Port Jefferson, New York, USA*, volume 1471 of *Lecture Notes in Computer Science*, pages 224–246. Springer, October 1998. ISBN 3-540-64958-1.

[49] M. Lenzerini and D. F. Savo. On the evolution of the instance level of DL-Lite knowledge bases. In R. Rosati, S. Rudolph, and M. Zakharyaschev, editors, *Proceedings of the 24th International Workshop on Description Logics (DL 2011)*, volume 745 of *CEUR Workshop Proceedings*, Barcelona, Spain, July 13-16 2011. CEUR-WS.org.

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

[51] H. Liu, C. Lutz, M. Miličić, and F. Wolter. Updating description logic ABoxes. In P. Doherty, J. Mylopoulos, and C. A. Welty, editors, *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR'06)*, pages 46–56, Lake District of the United Kingdom, June 2-5 2006. AAAI Press. ISBN 978-1-57735-271-6.

[52] J. W. Lloyd. *Foundations of Logic Programming*. Springer, 2nd edition, 1987. ISBN 3-540-18199-7.

[53] V. W. Marek and M. Truszczynski. Revision programming. *Theoretical Computer Science*, 190(2):241–277, 1998.

[54] B. Motik and R. Rosati. Reconciling description logics and rules. *Journal of the ACM*, 57(5):93–154, 2010.

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

[56] D. Pearce. A new logical characterisation of stable models and answer sets. In J. Dix, L. M. Pereira, and T. C. Przymusinski, editors, *Proceedings of the 6th Workshop on Non-Monotonic Extensions of Logic Programming (NMELP '96)*, volume 1216 of *Lecture Notes in Computer Science*, pages 57–70, Bad Honnef, Germany, September 5-6 1997. Springer. ISBN 3-540-62843-6.

[57] G. Qi and F. Yang. A survey of revision approaches in description logics. In D. Calvanese and G. Lausen, editors, *Proceedings of the 2nd International Conference on Web Reasoning and Rule Systems (RR'08)*, volume 5341 of *Lecture Notes in Computer Science*, pages 74–88, Karlsruhe, Germany, October 31-November 1 2008. Springer Verlag. ISBN 978-3-540-88736-2.

[58] M. M. Ribeiro and R. Wassermann. Base revision in description logics - preliminary results. In *Proceedings of the International Workshop on Ontology Dynamics (IWOD-07)*, pages 69–82, 2007.

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

[60] C. Sakama and K. Inoue. An abductive framework for computing knowledge base updates. *Theory and Practice of Logic Programming (TPLP)*, 3(6):671–713, 2003.

[61] J. Šefránek. Irrelevant updates and nonmonotonic assumptions. In M. Fisher, W. van der Hoek, B. Konev, and A. Lisitsa, editors, *Proceedings of the 10th European Conference on Logics in Artificial Intelligence (JELIA 2006)*, volume 4160 of *Lecture Notes in Computer Science*, pages 426–438, Liverpool, UK, September 13-15 2006. Springer. ISBN 3-540-39625-X.

[62] J. Šefránek. Static and dynamic semantics: Preliminary report. *Mexican International Conference on Artificial Intelligence*,

pages 36–42, 2011.

[63] J. Siska. Dynamic logic programming and world state evaluation in computer games. In M. Fink, H. Tompits, and S. Woltran, editors, *Proceedings of the 20th Workshop on Logic Programming*, volume 1843-06-02 of *INFSYS Research Report*, pages 64–70, Vienna, Austria, February 22-24 2006. Technische Universität Wien, Austria.

[64] M. Slota and J. Leite. Towards Closed World Reasoning in Dynamic Open Worlds. *Theory and Practice of Logic Programming, 26th Int'l. Conference on Logic Programming (ICLP'10) Special Issue*, 10(4-6):547–564, July 2010.

[65] M. Slota and J. Leite. Back and forth between rules and SE-models. In J. P. Delgrande and W. Faber, editors, *Proceedings of the 11th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-11)*, volume 6645 of *Lecture Notes in Computer Science*, pages 174–186, Vancouver, Canada, May 16-19 2011. Springer. ISBN 978-3-642-20894-2.

[66] M. Slota and J. Leite. Robust equivalence models for semantic updates of answer-set programs. In G. Brewka, T. Eiter, and S. A. McIlraith, editors, *Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR 2012)*, pages 158–168, Rome, Italy, June 10-14 2012. AAAI Press. ISBN 978-1-57735-560-1.

[67] M. Slota and J. Leite. A unifying perspective on knowledge updates. In L. F. del Cerro, A. Herzig, and J. Mengin, editors, *Proceedings of the 13th European Conference on Logics in Artificial Intelligence (JELIA 2012)*, volume 7519 of *Logics in Artificial Intelligence (LNAI)*, pages 372–384, Toulouse, France, September 26-28 2012. Springer.

[68] M. Slota and J. Leite. On condensing a sequence of updates in answer-set programming. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13)*. AAAI Press, 2013. To appear.

[69] M. Slota and J. Leite. The rise and fall of semantic rule updates based on SE-models. *Theory and Practice of Logic Programming (TPLP)*, 2013. To appear.

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

[71] H. Turner. Strong equivalence made easy: nested expressions and weight constraints. *Theory and Practice of Logic Programming (TPLP)*, 3(4-5):609–622, 2003.

[72] M. Winslett. Reasoning about action using a possible models approach. In *Proceedings of the 7th National Conference on Artificial Intelligence (AAAI 1988)*, pages 89–93, Saint Paul, MN, USA, August 21-26 1988. AAAI Press / The MIT Press. ISBN 0-262-51055-3.

[73] M. Winslett. *Updating Logical Databases*. Cambridge University Press, New York, USA, 1990. ISBN 0-521-37371-9.

[74] K.-S. Wong. A stronger notion of equivalence for logic programs. In V. Dahl and I. Niemelä, editors, *Proceedings of the 23rd International Conference on Logic Programming (ICLP 2007)*, volume 4670 of *Lecture Notes in Computer Science*, pages 453–454, Porto, Portugal, September 8-13 2007. Springer.

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