On the Conservativity and Stability of Ontology-Revision Operators Based on Reinterpretation

Özgür L. Özçep and Carola Eschenbach

Department for Informatics (AB WSV), MIN Faculty, University of Hamburg
Vogt-Kölln-Str. 30, D-22527 Hamburg, Germany
{oezcep,eschenbach}@informatik.uni-hamburg.de

Abstract. This article deals with the problem of integrating possibly conflicting information into an ontology. We define and analyze a family of ontology-revision operators that resolve conflicts by reinterpreting concept symbols occurring in the triggering information. The analysis of the iterated application of the operators focusses on issues of conservativity and stability of the ontology extension.

1 Introduction

Communication between natural or artificial agents relies on using shared terms with shared meanings. This precondition, however, cannot always be established in advance. While human users of natural language have flexible means to handle situations where different uses of the same term become obvious, such mechanisms of reinterpretations are not well studied for artificial agents. In this article, we are concerned with the specific case of heterogeneity between terminologies, where different agents use the same term with different meanings (conf. [1]) and where this ambiguity is discovered while the sender agent gives information that conflicts with the information the receiver holds. The approach aims at handling the communication between agents that hold kindred ontologies where conflicts are the exception rather than the rule. Therefore no preprocessing stage of aligning the terminologies in advance is assumed.

The specification of the terminology used in communication is based on an ontology the agent holds. For agents whose ontologies are consistent and well-tried the treatment of observed heterogeneities should not lead to the loss of (parts of) its former ontology. Therefore, we are faced with the problem of establishing a semantic mapping between the receiver's (internal) ontology and the sender's terminology during the exchange of information using the terms rather than the exchange of information about the terminologies. In this article we will focus on a lifting process where the incoming information is handled as a sequence of facts and the ontology of the sender is not communicated.

We outline the theoretical basis on which to generate semantic mappings as the product of applying a consistency resolving change operator to an ontology, represent semantic mappings as description logical formulas in the object language and use them like other logical formulas as premises for inferences needed to calculate the outcome of the change operators.

The ontology-revision operators defined and analyzed in this article are motivated by ideas from the area of belief revision. Along with a treatment of iterated revision (iterated application of an revision operator), we will discuss stability aspects for the operators.

A concrete application of the analyzed ontology-revision operators could be to embed them into an information processing system IPS. More concretely imagine a software agent that holds an ontology O_R. The IPS formulates a query (e.g., 'List all cheap books on thermodynamics') and sends it as a request to another agent (the sender) that offers services concerning the request, e.g., services that are needed for online book stores. The sender processes the request, generates a response by using its own ontology O_S, and sends the response as a sequence of information. The IPS processes the sequence by applying the revision operator (incrementally) and thereby resolves conflicts that possibly occur due to the difference between O_R and O_S, thereby, e.g., discovering that the concept *cheap* has different meanings in O_S and O_R.

2 Ontology-revision Operators: Definitions

Following M. Grove's idea of so called sphere-based belief revision outlined in [2], Wassermann/Fermé ([3]) constructed operators for expanding, revising and contracting a set of concept descriptions by a concept description. As ontologies deal with concepts, [4] adapted these ideas in order to define ontology-revision operators that get as input an ontology O and a sentence α, also called the trigger information, and that have as output a new ontology. Two different types of operators $\odot_1$ and $\odot_2$ were defined in a local and global variant respectively. In this article only the global variants will be dealt with.

For the definition of the operators some preliminary notation is necessary. Throughout this article an ontology will be understood as a finite set of sentences over a description logical (DL) language. We will use the DL syntax to describe ontologies and the trigger information, no special DL will be assumed to be given in advance.

An ontology will be denoted by O and indexed or primed variants. An *ontology over a language* $\mathcal{L}$ is a set of sentences in which all non-logical symbols, i.e., the concept symbols, constants and role symbols, are among those in $\mathcal{L}$. $\mathcal{L}(O)$ describes exactly the non-logical symbols occurring in O. Writing $\alpha \in \mathcal{L}$ for a sentence α means that all non-logical symbols of α are in $\mathcal{L}$. Concept descriptions will be denoted by C and indexed or primed variants. Concept symbols (or atomic concept descriptions) will be denoted by K, S, T and indexed or primed variants. Constants will be denoted by $a, b, c \ldots$ and indexed variants. $O_{[K_1/K_2]}$ is the outcome of uniformly substituting K_1 by K_2 in O. Sentences of the form $K(a)$ (for K a concept symbol) will be called positive literals, sentences of the form $\neg K(a)$ will be named negative literals and the union of these sets of sentences will be simply named literals. $\mathsf{Mod}(O)$ is the set of models of O.

The (global) operators of [4] are defined with reference to the most specific concept assigned by an ontology to a constant. C is a most specific concept (msc) for a in the ontology O iff $O \models C(a)$ and for all C' such that $O \models C'(a)$ it is true that $O \models C \sqsubseteq C'$. The existence of a most specific concept depends on the ontology O and the underlying description logic.[1] We assume that there is some systematic way (e.g. an ordering over concept descriptions) to pick out for every constant a a unique most specific concept in an ontology O. This unique most specific concept will be denoted by $\mathsf{msc}_O(a)$ and we will talk about *the* most specific concept of a constant in an ontology (or regarding an ontology).

Definition 1. *Let O be an ontology over a DL-Language $\mathcal{L}$, $\alpha = K(a)$ a sentence in $\mathcal{L}$ with K a concept symbol and let a be a constant for which $\mathsf{msc}_O(a)$ exists. Let K' be a new concept symbol not occurring in $O \cup \{\alpha\}$. Then the **global operators of type 1 and 2 (for positive literals)** are defined resp. by*[2]

$$O \odot_1 K(a) = \begin{cases} O \cup \{K(a)\} & \textit{if } O \cup \{K(a)\} \textit{ is consistent,} \\ O \cup \{K \sqsubseteq K', K' \sqsubseteq K \sqcup \mathsf{msc}_O(a), \\ \qquad K'(a)\} & \textit{else} \end{cases} \tag{1}$$

$$O \odot_2 K(a) = \begin{cases} O \cup \{K(a)\} & \textit{if } O \cup \{K(a)\} \textit{ is consistent,} \\ O_{[K/K']} \cup \{K' \sqsubseteq K, K \sqsubseteq K' \sqcup \mathsf{msc}_{O_{[K/K']}}(a), \\ \qquad K(a)\} & \textit{else} \end{cases} \tag{2}$$

The operators $\odot_1$ and $\odot_2$ are similar as one can obtain $\odot_2$ by changing the roles of K' and K in the definition of $\odot_1$. The case that the union of the trigger information α and the ontology O is consistent is handled by both operators in the same way by adding α to O. If $O \cup \{\alpha\}$ is inconsistent, the inconsistency is ascribed to the fact that the occurrences of K in O and in α respectively have different meanings, i.e., denote different concepts. This difference in the meanings is represented via disambiguating K by introducing a new symbol K' that denotes the other concept.

The operators $\odot_1$ and $\odot_2$ differ regarding which concept (the concept represented in O vs. the concept represented in α) is denoted by the new symbol K'. The type-1 operator $\odot_1$ substitutes the occurrence of K in the trigger information by a new symbol K' while O is not changed. We will also say that K in α is *reinterpreted*. The type-2 operator $\odot_2$ substitutes the occurrences of K in the ontology O by a new symbol K', while preserving α. We will also say that K in O is reinterpreted. The difference between $\odot_1$ and $\odot_2$ can also be described by saying that $\odot_1$ preserves the terminology of the ontology O while $\odot_2$ adapts to the terminology of the sender of α.

As a consequence, O is not changed by applying $\odot_1$ and α is put in a reinterpreted form into the resulting ontology. The operator $\odot_1$ fulfills the condition

[1] [5] describes a family of description logics for which the most specific concept exists and an algorithm for determining the most specific concept.

[2] [4], p. 87

of monotonicity (see below) but only a weak form of the success postulate mentioned in the classic belief revision postulates of AGM.[3] The operator $\odot_2$, on the other hand, fulfills the success axiom, i.e., $\alpha \in O \odot_2 \alpha$, but not monotonicity.

Both operators declare upper and lower bounds in which the old symbol K and the new symbol K' occur. In case of $\odot_1$ the bounds are given for K' depending on K and in case of $\odot_2$ the bounds are given for K depending on K'.

A limitation of the definitions for $\odot_1$ and $\odot_2$ is the fact that they deal only with positive literals. In order to widen the applicability of the operators, we extend the definitions of the operators to deal also with negative literals.[4]

Definition 2. *Let O be an ontology over a DL-Language $\mathcal{L}$, K a concept symbol, and let a be a constant in $\mathcal{L}$ for which $\mathsf{msc}_O(a)$ exists. Let K' be a new concept symbol not occurring in $O \cup \{\neg K(a)\}$. Then the* ***global operators of type 1 and 2 (for literals)*** *are defined according to Def. 1 for the positive cases and for the negative cases by*

$$O \odot_1 \neg K(a) = \begin{cases} O \cup \{\neg K(a)\} & \textit{if } O \cup \{\neg K(a)\} \textit{ is consistent,} \\ O \cup \{K' \sqsubseteq K, K \sqcap \neg \mathsf{msc}_O(a) \sqsubseteq K', & \\ \qquad \neg K'(a)\} & \textit{else} \end{cases} \tag{3}$$

$$O \odot_2 \neg K(a) = \begin{cases} O \cup \{\neg K(a)\} & \textit{if } O \cup \{\neg K(a)\} \textit{ is consistent,} \\ O_{[K/K']} \cup \{K \sqsubseteq K', K' \sqcap \neg \mathsf{msc}_{O_{[K/K']}}(a) \sqsubseteq K, & \\ \qquad \neg K(a)\} & \textit{else} \end{cases} \tag{4}$$

The use of most specific concepts in the definitions results from the construction in [4], in which the global operators (defined above) originate as generalizations of the local operators—using the most specific concept as a common bound for all local operators.

By weakening the specification, yielding definitions for the operators $\otimes_1$ and $\otimes_2$, we get rid of the reference to most specific concepts. The analysis of the operators $\otimes_1$ and $\otimes_2$ aims at preparing the analysis of the stronger operators $\odot_1$ and $\odot_2$ and will show that these weak operators also have some undesirable properties.

Definition 3. *Let O be an ontology over a DL-Language $\mathcal{L}$, K a concept symbol and let a be a constant in $\mathcal{L}$. Let K' be a new concept symbol not occurring in $O \cup \{K(a)\}$. Then the* ***weak global operators of type 1 and 2 (for literals)*** *are defined by*

$$O \otimes_1 K(a) = \begin{cases} O \cup \{K(a)\} & \textit{if } O \cup \{K(a)\} \textit{ is consistent,} \\ O \cup \{K \sqsubseteq K', K'(a)\}, & \textit{else} \end{cases} \tag{5}$$

$$O \otimes_1 \neg K(a) = \begin{cases} O \cup \{\neg K(a)\} & \textit{if } O \cup \{\neg K(a)\} \textit{ is consistent,} \\ O \cup \{K' \sqsubseteq K, \neg K'(a)\} & \textit{else} \end{cases} \tag{6}$$

[3] [6], p. 513

[4] The extension of the definitions to other types of trigger information is more complex since it needs to handle more than one candidate for reinterpretation.

$$O \otimes_2 K(a) = \begin{cases} O \cup \{K(a)\} & \textit{if } O \cup \{K(a)\} \textit{ is consistent,} \\ O_{[K/K']} \cup \{K' \sqsubseteq K, K(a)\} & \textit{else} \end{cases} \quad (7)$$

$$O \otimes_2 \neg K(a) = \begin{cases} O \cup \{\neg K(a)\} & \textit{if } O \cup \{\neg K(a)\} \textit{ is consistent,} \\ O_{[K/K']} \cup \{K \sqsubseteq K', \neg K(a)\} & \textit{else} \end{cases} \quad (8)$$

The operators $\odot_i$ and $\otimes_i$ for ($i \in \{1, 2\}$) can be considered as special cases of the operators $\oplus_i^{\mathsf{sel}}$ defined (for positive literals) according to (9) and (10):

$$O \oplus_1^{\mathsf{sel}} K(a) = O \otimes_1 K(a) \cup \{K \sqsubseteq K' \sqcup \mathsf{sel}(\{C \mid O \models C(a)\})\} \quad (9)$$

$$O \oplus_2^{\mathsf{sel}} K(a) = O \otimes_2 K(a) \cup \{K' \sqsubseteq K \sqcup \big(\mathsf{sel}(\{C \mid O \models C(a)\})\big)_{[K/K']}\} \quad (10)$$

The operator $\oplus_i^{\mathsf{sel}}$ has a selection function sel as a parameter that, in order to warrant consistency, selects one concept $M = \mathsf{sel}(\{C \mid O \models C(a)\})$ from the set of concepts C instantiated by a in O. If sel is such that $M = \top$, one gets the operator $\otimes_2$. If sel is such that $M = \mathsf{msc}_{\mathsf{O}}(a)$, one gets the operator $\odot_2$.[5]

In this article we will focus on the operators $\odot_i$ and $\otimes_i$ for ($i \in \{1, 2\}$) thereby avoiding the additional complexity due to the selection function sel.

One of the main questions of this article is how the operators behave in case a finite sequence of literals $A = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ or an infinite sequence of trigger information is to be integrated into an ontology. To formulate this question, we use some additional notation: Let $\circ \in \{\odot_1, \odot_2, \otimes_1, \otimes_2\}$ be an operator, $A = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ a finite sequence of literals. Then $O \circ A =_{\text{def}} (\ldots(O \circ \alpha_1) \circ \alpha_2) \ldots) \circ \alpha_n$ describes the outcome of iterated applications of the operator $\circ$ to the resulting ontologies and the trigger information of the sequence A. In case the sequence A is known and has length n we will use $O^{\circ(n)}$ instead of $O \circ A$ and even shorter $O^{(n)}$ if it is clear from the context which operator is meant (or if it is not relevant for which operator repeated application is considered). If $A = (\alpha_1, \ldots, \alpha_i, \ldots)$, then let $O^{\circ(i)} = (\ldots(O \circ \alpha_1) \circ \ldots \circ \alpha_i)$. If A is a sequence of length n, then A^i (for $i \leq n$) is the prefix of A of length i.

For simplicity we will treat sequences of literals like sets of the literals whenever the order is irrelevant. Thus we write $\alpha \in A$ or $A \subseteq O$ or $O \cup A$. The symbol '$\circ_1$' will be used as metavariable for type-1 operators, i.e., '$\circ_1$' stands for $\odot_1$ or $\otimes_1$, and '$\circ_2$' will be used as metavariable for type-2 operators, i.e., '$\circ_2$' stands for $\odot_2$ or $\otimes_2$.

3 Monotonicity and Non-monotonicity

A simple observation directly resulting from the definitions of the operators is the following:

Observation 1. *Let O be an ontology over $\mathcal{L}$, $\alpha \in \mathcal{L}$ and A a sequence of literals. Let $\circ_1$ be a type-1 operator and $\circ_2$ be a type-2 operator. Then:*

[5] The selection function sel has a role similar to the role of the selection functions defined in [6] for use in partial meet revision and its special cases maxi-choice and full meet revision.

1. *$O \subseteq O \circ_1 \alpha$ (monotonicity of $\circ_1$)*
2. *For all $n \in \mathbb{N}$: $O \subseteq O^{\circ_1(n)}$ (monotonicity of iterated $\circ_1$)*
3. *$O \otimes_i \alpha \subseteq O \odot_i \alpha$, for $i \in \{1, 2\}$ ($\odot_i$ is at least as strong as $\otimes_i$)*
4. *$\alpha \in O \circ_2 \alpha$ (success for $\circ_2$)*
5. *$O \circ \alpha = O \cup \{\alpha\}$ iff $O \cup \{\alpha\}$ is consistent.*
6. *$O \subseteq O \circ A = O \cup A$ iff $O \cup A$ is consistent.*
7. *$O \circ \alpha$ is consistent.*[6]
8. *If $\mathsf{Mod}(O_1) = \mathsf{Mod}(O_2)$, then $\mathsf{Mod}(O_1 \circ \alpha) = \mathsf{Mod}(O_2 \circ \alpha)$ (syntax independence)*
9. *If $O \cup \{\alpha\}$ is inconsistent and K' is the new symbol introduced in $O \circ_2 \alpha$ resp. $O \circ_1 \alpha$, then: $(O \circ_2 \alpha)_{[K/L, K'/K]} = (O \circ_1 \alpha)_{[K'/L]}$, for $L \neq K' \notin \mathcal{L}(O \cup \{\alpha\})$ and $\alpha = K(a)$ or $\alpha = \neg K(a)$.*

Assertions 1.4,1.5, 1.7 and 1.8 of the observation are four adapted variants from six of the AGM postulates.[7] The other two postulates deal with the revision of belief sets/propositions with complex information which we cannot (yet) simulate in our setting as we defined the operators only for literals.

While type-1 operators are monotone, type-2 operators can be non-monotone, e.g. $O \not\subseteq O \circ_2 \alpha$ if $O \cup \{\alpha\}$ is inconsistent.

In the case of inconsistency, one can say a little bit more about the behavior of type-1 operators: The integration of α into O results in a conservative extension. According to the usual logical use of the term a theory O' in a language $\mathcal{L}'$ is called a *conservative extension of the theory* O in a language $\mathcal{L} \subseteq \mathcal{L}'$ iff for all sentences α in $\mathcal{L}$: $O \models \alpha$ iff $O' \models \alpha$.[8] The following proposition states conservativity:

Proposition 1. *Let O be an ontology over a language $\mathcal{L}$, and $\alpha \in \mathcal{L}$ be a literal. Then: If $O \cup \{\alpha\}$ is inconsistent, then $O \odot_1 \alpha$ and $O \otimes_1 \alpha$ are conservative extensions of O.*

Proof. See p. 13.

In the consistency case one cannot guarantee $O \circ_1 \beta$ to be a conservative extension, only the property of monotonicity holds. As a consequence it is not the case that for all n: $O^{\circ_1(n)}$ is a conservative extension of O. Additionally the following observations can be made:

Observation 2. *For an ontology O over $\mathcal{L}$ and literals $\alpha, \alpha_i, \alpha_j \in \mathcal{L}$:*

1. *The outcome of applying $\circ_1$ to a sequence A of literals depends on the order of the elements in A. In case of $O \models \neg(\alpha_i \wedge \alpha_j)$ for $\alpha_i, \alpha_j \in A$ and $i < j$ it is possible that α_i wins/survives when resolving the conflict in step j.*
2. *There is a subset A', such that: $O \cup A' \subseteq O^{\circ_1(n)}$ and $O^{\circ_1(n)}$ is a conservative extension of $O \cup A'$.*

[6] This can be proved as a corollary to Prop. 1.
[7] Compare the (re-)formulation of the postulates in [7].
[8] [8], p. 208 and [9], p. 625.

3. *The monotonicity of* $\circ_1$ *preserves conflicts: If* $O \cup \{\alpha\}$ *is inconsistent, then* $O^{(n)} \cup \{\alpha\}$ *is also inconsistent. Thus, if* $O \cup \{\alpha\}$ *is inconsistent, repeated occurrences of* α *in* A *never result in* $O^{(n)} \models \alpha$ *for any* $n \in \mathbb{N}$.

The operators of type 2 are not monotone. Therefore the analysis of the type-2 operators is more complicate. But in combination with the fact that the success condition is fulfilled stability (in a intuitive sense defined below) is provable (at least for the weak type-2 operators). From now on we will concentrate on type-2 operators.

4 Detailed Analysis

4.1 Restricted Conservativity

The following proposition states restricted conservativity properties for the operators $\odot_2$ and $\otimes_2$. More precisely, (14) and (18) state conservativity for all sentences β that do not contain one of the concept symbols K, K' (directly) involved in the reinterpretation. Assertions (11) and (15) express conservativity for those sentences that are literals and in which the reinterpreted symbol K occurs with the same prefix (negation vs. no negation symbol) as in the triggering information. Similarly (12) and (16) express conservativity (only in case of the strong operator $\odot_2$) for those sentences that are literals and in which the reinterpreted symbol K occurs with a different (complementary) prefix as in the triggering information. Assertions (13) and (17) express the fact that the weak operator $\otimes_2$ does not preserve literals in which the reinterpreted symbol K occurs with a different prefix than the prefix of the occurrence of K in the trigger information.

Proposition 2. *Let* a *and* c *be constants,* K *be a concept symbol,* O *be an ontology such that* $\mathsf{msc_O}(a)$ *exists. Let* $\mathcal{L} = \mathcal{L}(O \cup \{K(a), K(c)\})$. *Then for all formula* $\beta \in \mathcal{L} \setminus \{K, K'\}$:

– *If* $O \models \neg K(a)$, *then:*

$$O \circ_2 K(a) \models K(c) \ \text{ iff } \ O \cup \{a \neq c\} \models K(c) \tag{11}$$
$$O \odot_2 K(a) \models \neg K(c) \ \text{ iff } \ O \models \neg K(c) \text{ and } O \models \neg \mathsf{msc_O}(a)(c) \tag{12}$$
$$O \otimes_2 K(a) \not\models \neg K(c) \tag{13}$$
$$O \circ_2 K(a) \models \beta \ \text{ iff } \ O \models \beta \tag{14}$$

– *If* $O \models K(a)$, *then:*

$$O \circ_2 \neg K(a) \models \neg K(c) \ \text{ iff } \ O \cup \{a \neq c\} \models \neg K(c) \tag{15}$$
$$O \odot_2 \neg K(a) \models K(c) \ \text{ iff } \ O \models K(c) \text{ and } O \models \neg \mathsf{msc_O}(a)(c) \tag{16}$$
$$O \otimes_2 \neg K(a) \not\models K(c) \tag{17}$$
$$O \circ_2 \neg K(a) \models \beta \ \text{ iff } \ O \models \beta \tag{18}$$

Proof. See p. 13.

The operators $\odot_2$ and $\otimes_2$ nearly fulfill the same restricted conservativity assertions. The crucial difference is expressed by (12) and (13) (for positive literals) and (16) and (17) (for negative literals). Because of this we can infer more about $\otimes_2$ than is expressed in Prop. 2. This will be stated in Sect. 4.2 on stability.

The conservativity properties expressed in Prop. 2 are called 'restricted' because of two reasons: 1) Conservativity holds only for a subset of the sentences (the set of literals) and 2) the 'if'-directions of two of the proposed assertions ((11), (15)) hold only with additional assumptions concerning the uniqueness of constants. These additional assumptions will be called *'local unique name assumptions'* and abbreviated by 'UNA'. They express for some (not all) constants occurring in the ontology and the trigger information the condition that they denote different entities.[9]

The local unique name assumptions have a crucial role in the question of stability which we deal with in the next subsection.

4.2 Stability

The main setting we consider is that of an agent holding some ontology O and receiving a sequence A of trigger information (all being literals) and integrating them into its ontology by using an operator of type 2. If the trigger information stems from the same source ontology and this ontology is consistent, also A is consistent. We focus on cases for which A contains only a finite set of different literals and for which some literals can occur infinitely often in A. As the operators of type 2 fulfill success the question arises whether there is a step during integrating A from which on the ontology does not change anymore. Formally asked, the question is: Is there some $i \in \mathbb{N}$ such that $O^{\circ_2(i+m)} = O^{\circ_2(i)}$ for all $m \in \mathbb{N}$?

For the weak operator $\otimes_2$ stability holds under some local unique name assumptions. Stability in general does not hold without a (local) UNA. This can be demonstrated by a simple example.

Example 1. Consider the ontology $O = \{R(c,a), (\leq 1R)(c), R(c,b)\}$. Then $O \models a = b$. If A is the infinite sequence $(K(a), \neg K(b), K(a), \neg K(b), \ldots)$ (having finite different literals), then stability cannot occur. In other words: If according to the ontology of the receiver one object is denoted by two different constants a, b but according to the ontology of the sender a, b denote different objects, then this mismatch cannot be solved by an operator of type 2.

The stability question for $\odot_2$ is a bit more complex because of the additional bound containing the most specific concept. The problematic fact in case of $\odot_2$

[9] It is not enough to replace the UNA $a \neq c$ in (11) and (15) by $O \not\models a = c$. The following, e.g., is not true in general: $O \not\models a = c$ and $O \models K(c)$ iff $O \otimes_2 K(a) \models K(c)$. A counterexample is given by $O = \{\neg K \equiv \mathsf{one{-}of}(\{a\})\}$.

is that information integrated in one step i may disappear in a later step $i+m$ and perhaps be replaced by its negation in another (or the same) step. This is demonstrated by the following example.

Example 2. Let the ontology O and the sequence A be given by

$$\begin{aligned} O &= \{\neg K(a_1), L(a_1), L(a_2)\} \\ A &= (\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (\neg K(a_2), K(a_1), \neg L(a_1), \neg L(a_2)) \end{aligned}$$

Applying the definition of $\odot_2$ results in

$$\begin{aligned} O \odot_2 A \equiv \{&\neg K'(a_1), L'(a_1), L'(a_2), \neg K'(a_2), K(a_1), K' \sqsubseteq K, \\ &K \sqsubseteq K' \sqcup L', \neg L(a_1), L \sqsubseteq L', L' \sqcap (\neg K' \sqcup \neg K)) \sqsubseteq L, \neg L(a_2)\} \end{aligned}$$

Consequently $O \odot_2 A \models \neg\alpha_1$, i.e., the trigger information α_1 from the first step is abandoned in a later step and its negation follows from $O \odot_2 A$. Thus success for α_1 is not warranted.

This example does not show that stability cannot hold for $\odot_2$, but it shows that we cannot prove it by proving $O^{(i)} \not\models \neg\alpha_1$.

The main result of this article is the stability of $\otimes_2$ and can be proved as a corollary to the following theorem. The unique name assumption used in the theorem is defined for a sequence A by

$$\mathsf{una}(A) = \{a \neq b \mid K(a), \neg K(b) \in A, \text{ for a concept symbol } K\} \ .$$

Defining $\mathsf{una}(A)$ in this way, also expresses the assumption that the set of literals in A is consistent.

Theorem 1. *Let O be a consistent ontology over $\mathcal{L}$. Then for all finite sequences A of literals in $\mathcal{L}$:*
If $(O \otimes_2 A) \cup \mathsf{una}(A)$ is consistent, then $(O \otimes_2 A) \cup \mathsf{una}(A) \cup A$ is consistent as well.

Sketch of Proof. We need some additional notation. For a sequence $A = (\alpha_i)_{i \in I}$ and a concept symbol K let

$$A_K = \{\alpha_j \mid j \in I, \alpha_j = K(a_j) \text{ for some constant } a_j\}$$

be the set of literals contained in A in which K occurs positively. Accordingly

$$A_{\neg K} = \{\alpha_j \mid j \in I, \alpha_j = \neg K(a_j) \text{ for some constant } a_j\}$$

is the set of literals contained in A in which K occurs negatively. Let $A_{(K)} = A_K \cup A_{\neg K}$. With $O_K = \{\beta \in O \mid \beta \text{ contains } K\}$ we describe that part of the ontology O that syntactically contains K. Let $\mathcal{K} = \{K_i \mid i \in I\}$ be the set of all concept symbols in $\mathcal{L}$ for some $I \subseteq \mathbb{N}$.

The main ideas in the proof outlined in the following are first to separate the ontologies in different parts according to the concept symbols and second to check the following two facts: 1) If a conflict resolution for a literal $\alpha_i = K(a)$ is done in step i, then a conflict resolution for a literal α_j (integrated in step $j > i$) containing the same concept symbol K can only occur if α_j has the form $\neg K(b)$. (Accordingly if $\alpha_i = \neg K(a)$, then α_j must have the form $K(b)$.) 2) There can be at most two conflict resolutions with respect to the same concept symbol.

The proof is done by induction on the length of the sequence A. In fact the assertion proved by induction is stronger than the one formulated in the theorem, and it contains the assertion of the theorem as (the last) conjunctive part. The assertion is:

For all finite sequences A of length n: There are two disjoint sets of concept symbols $\mathcal{K'}_n$ and $\mathcal{K''}_n$ that are disjoint from $\mathcal{K}$ and have the form $\mathcal{K}'_n = \{K'_i \mid i \in I'_n\}$ and $\mathcal{K}''_n = \{K''_i \mid i \in I''_n\}$ for $I''_n \subseteq I'_n \subseteq I$. And there is a substitution σ_n defined by $K_i\sigma_n = K'_i$ if $i \in I'_n$ and $K_i\sigma_n = K_i$ else, such that the following five assertions hold:

1. $O\sigma_n \subseteq O^{(n)}$
   This expresses the fact that the (original) ontology O in some way is preserved along the integration. It can be found in the resulting ontology $O^{(n)}$ by applying the substitution σ_n which maps the concept symbols of the old ontology onto the corresponding (primed new) symbols of the new ontology and thereby acts like a semantic mapping.
2. All concept symbols contained in $O^{(n)}$ are contained in $\mathcal{K} \cup \mathcal{K}'_n \cup \mathcal{K}''_n$.
3. $O^{(n)}$ can be represented by

$$
\begin{aligned}
O^{(n)} = O\sigma_n \cup \overbrace{\bigcup_{i \in I \setminus I'_n} A_{(K_i)}}^{\text{no revision}} \cup \overbrace{\bigcup_{i \in I'_n \setminus I''_n} \left(O^{(n)}_{K_i} \cup (A_{(K_i)} \setminus O^{(n)}_{K_i})\sigma_n\right)}^{\text{simple revision}} \cup \\
\underbrace{\bigcup_{i \in I''_n} \left(O^{(n)}_{K_i} \cup O^{(n)}_{K''_i} \cup \left(A_{(K_i)} \setminus O^{(n)}_{K_i} \setminus O^{(n)}_{K''_i[K''_i/K_i]}\right)\sigma_n\right)}_{\text{twofold revision}}
\end{aligned}
$$

   As the comments under and over the cambered brackets suggest, there can be maximally two conflict resolutions with respect to the same concept symbol.
4. For all $i \in I$:
   (a) If $i \notin I'_n$, then $A_{(K_i)} \subseteq O^{(n)}$.
   (b) If $i \in I'_n \setminus I''_n$, then $(A_{(K_i)} \setminus O^{(n)}_{K_i})\sigma_n \subseteq O^{(n)}$ and there is exactly one T-box axiom in $O^{(n)}$ of the form $K'_i \sqsubseteq K_i$ (case (A)) or $K_i \sqsubseteq K'_i$ (case (B)).
   (A) In this case additionally
   $O^{(n)}_{K_i} \subseteq \{K'_i \sqsubseteq K_i\} \cup A_{(K_i)}$ and if $O^{(n)} \models \neg K_i(a_j)$, then $K_i(a_j) \notin A$.
   (B) In this case additionally
   $O^{(n)}_{K_i} \subseteq \{K_i \sqsubseteq K'_i\} \cup A_{(K_i)}$ and if $O^{(n)} \models K_i(a_j)$, then $\neg K_i(a_j) \notin A$.
   (c) If $i \in I''_n$, then there is $K''_i \in \mathcal{K}''_n$ such that $(A_{(K_i)} \setminus O^{(n)} \setminus O^{(n)}_{K''_i[K''_i/K_i]})\sigma_n \subseteq O^{(n)}$ and there is exactly one T-box axiom of the form $K''_i \sqsubseteq K_i$ (case (A)) or of the form $K_i \sqsubseteq K''_i$ (case (B)).

(A) In this case additionally
- $K_i'' \sqsubseteq K_i' \in O^{(n)}$ and $O_{K_i}^{(n)} \subseteq \{K_i'' \sqsubseteq K_i\} \cup A_{(K_i)}$
- $O_{K_i''}^{(n)} \subseteq \{K_i'' \sqsubseteq K_i', K_i'' \sqsubseteq K_i'\} \cup (A_{(K_i)})_{[K_i/K_i'']}$
- $O^{(n)} \models (A_{\neg K_i})_{[K_i/K_i'']} \cup (A_{K_i} \cap O_{K_i}^{(n)}) \cup (A_{K_i} \setminus O_{K_i}^{(n)})\sigma_n$
- If $O^{(n)} \models \neg K_i(a_j)$, then $K_i(a_j) \notin A$ and
- if $O^{(n)} \models K_i(a_j)$, then $\neg K_i(a_j) \notin A$.

(B) In this case additionally
- $K_i' \sqsubseteq K_i'' \in O^{(n)}$ and $O_{K_i}^{(n)} \subseteq \{K_i \sqsubseteq K_i''\} \cup A_{(K_i)}$
- $O_{K_i''}^{(n)} \subseteq \{K_i \sqsubseteq K_i'', K_i' \sqsubseteq K_i''\} \cup (A_{(K_i)})_{[K_i/K_i'']}$
- $O^{(n)} \models (A_{K_i})_{[K_i/K_i'']} \cup (A_{\neg K_i} \cap O_{K_i}^{(n)}) \cup (A_{\neg K_i} \setminus O_{K_i}^{(n)})\sigma_n$
- If $O^{(n)} \models \neg K_i(a_j)$, then $K_i(a_j) \notin A$ and
- if $O^{(n)} \models K_i(a_j)$, then $\neg K_i(a_j) \notin A$.

5. If $(O \otimes_2 A) \cup \mathsf{una}(A)$ is consistent, then also $(O \otimes_2 A) \cup \mathsf{una}(A) \cup A$ is consistent.

The proof of the 5th assertion relies on the assertions before and is done by a model construction which completes the proof. Let $\mathcal{M}$ be a model of $(O \otimes_2 A) \cup \mathsf{una}(A)$. We construct a model $\mathcal{M}'$ of $(O \otimes_2 A) \cup \mathsf{una}(A) \cup A$ as follows:

- $\mathsf{dom}(\mathcal{M}') = \mathsf{dom}(\mathcal{M}) = D$; $\mathcal{M}'(a) = \mathcal{M}(a)$ for all constants a;
- $\mathcal{M}'(R) = \mathcal{M}(R)$ for all role symbols R; $\mathcal{M}'(K_i') = \mathcal{M}(K_i')$ for all $i \in I_n'$

$$- \ \mathcal{M}'(K_i) = \begin{cases} \mathcal{M}(K_i) & \text{if } i \notin I_n' \\ \mathcal{M}(K_i') \setminus \{\mathcal{M}(a_i) \mid \neg K_i(a_j) \in A\} & \text{if } i \in I_n' \setminus I_n'' \text{ and} \\ & K_i \sqsubseteq K_i' \in O^{(n)} \\ \mathcal{M}(K_i') \cup \{\mathcal{M}(a_i) \mid K_i(a_j) \in A\} & \text{if } i \in I_n' \setminus I_n'' \text{ and} \\ & K_i' \sqsubseteq K_i \in O^{(n)} \\ D \setminus \{\mathcal{M}(a_j) \mid \neg K_i(a_j) \in A\} & \text{if } i \in I_n'' \end{cases}$$

$$- \ \mathcal{M}'(K_i'') = \begin{cases} \mathcal{M}(K_i') \setminus \{\mathcal{M}(a_i) \mid \neg K_i(a_j) \in A\} & \text{if } K_i'' \sqsubseteq K_i' \in O^{(n)} \\ \mathcal{M}(K_i') \cup \{\mathcal{M}(a_i) \mid K_i(a_j) \in A\} & \text{if } K_i' \sqsubseteq K_i'' \in O^{(n)} \end{cases}$$

The theorem does not state success to be fulfilled with respect to a sequence A, i.e., it is not generally the case that $A \subseteq O \otimes_2 A$, but it states that a weakening of success is true in the sense that $O \otimes_2 A$ is at least compatible with A. But note that the interpretation of concept symbols that are subject to two revisions (K_i with $i \in I_n''$) solely depends on A and is completely independent of the original ontology. Therefore, further investigations on the behavior of the more complex operators of type 2 are called for.

As a corollary of the theorem the stability of $\otimes_2$ (in the sense mentioned above) results.

Corollary 1. *Let O be a consistent ontology and A an infinite sequence of literals containing a finite amount of different literals. Then if for all $j \in \mathbb{N}$ $O^{(j)} \cup \mathsf{una}(A)$ is consistent, there is a step $i \in \mathbb{N}$ such that*

$$O^{\otimes_2(i+m)} = O^{\otimes_2(i)} \text{ for all } m \in \mathbb{N}.$$

5 Related Work

Among the approaches that deal with belief-revision techniques to solve problems from the field of semantic integration, [10], [11] and especially [12] are most closely connected to our approach.

The idea of reinterpreting concepts is similar to the idea of weakening A-box axioms in [12] adapted from [10]. The authors of [12] describe revision operators for revising a consistent DL knowledge base KB by a another knowledge base KB' that contains at least one A-box axiom involved in the inconsistency. In the refined version of the revision operator, sentences of KB that are in conflict with those in KB' are replaced by some weakened versions. The leading idea behind the weakening strategy is to consider the cases that lead to the conflict as exceptions.

The main differences between [12] and our approach are that our conflict resolution is done by weakening a concept rather than by weakening sentences of the knowledge base. We focus on literals as triggering information whereby the construction of [12] handles knowledge bases consisting of more complex sentences. We consider iterated applications on a sequence of literals while [12] considers the revision with a set of sentences. Finally our conflict resolution involves a language extension that makes it possible to preserve the old ontology (knowledge base) and declare relations between the old and the new concepts.

6 Conclusion

The analysis of the type-2 operators yields restricted conservativity results and a stability theorem (for the weak version $\otimes_2$). The property of (restricted) conservativity in the inconsistency case is a form of informational conservativity as mentioned in the discussion of rationality postulates[10] for revision operators; this property offers the possibility to use the operators in those areas of information processing that include *refinement* as a main operation.[11]

The property of being stable makes the behavior of the (weak) operators of type 2 predictable. Coming back to the intended application scenario of an information processing system IPS with the embedded operator $\otimes_2$, this means that if we want a predictable behavior of the IPS, we should at least demand two conditions to be fulfilled in the scenario: 1) There should be only finitely many different literals in the sequence A of triggering literals. 2) The sequence A should be consistent. Scenarios in which both conditions are likely to be fulfilled are those in which A stems from a single sender whose knowledge base (ontology) is consistent. Scenarios in which A consists of trigger information from different senders consistency of A is likely not to be fulfilled. For those scenarios type-1 operators could be more appropriate than type-2 operators.

Theorem 1 only asserts compatibility of $O \otimes_2 A$ and A but not success for the whole sequence A (in the sense that $A \subseteq O \otimes_2 A$). This weakness could

[10] [13], p. 52–61.

[11] See [9] for a discussion of refinement.

be compensated by equipping an IPS with an additional memory in which all literals of A are stored and put into $O \otimes_2 A$ after the last literal of A was received.

Example 1 demonstrated the importance of (local) unique name assumptions without which stability is not warranted, and in fact the theorem presupposes the unique name assumption $\mathsf{una}(A)$. So a correctly working IPS would have to check the violation of the UNA and report it. (But this is not handled yet).

Appendix: Proofs

Proof of Prop. 1 (p. 6). Let $\circ_1 \in \{\odot_1, \otimes_1\}$ and $K' \notin \mathcal{L}$.

If β is a sentence in $\mathcal{L}$ and $O \models \beta$, then also $O \circ_1 \alpha \models \beta$, because $O \subseteq O \circ_1 \alpha$. Now suppose that $O \not\models \beta$ for $\beta \in \mathcal{L}$. We show the proposition for positive literals $\alpha = K(a)$. We have to show that $O \circ_1 K(a) \not\models \beta$. By assumption, there is a model $\mathcal{M} \models O \cup \neg\beta$ over $\mathcal{L}$. Define $\mathcal{M}'$ for the language $\mathcal{L}' = \mathcal{L} \cup \{K'\}$ as an extension of $\mathcal{M}$ with $\mathsf{dom}(\mathcal{M}) = \mathsf{dom}(\mathcal{M}')$, $\mathcal{M}'(S) = \mathcal{M}(S)$ for all symbols S different from K' and $\mathcal{M}'(K') = \mathcal{M}(K) \cup \{\mathcal{M}(a)\}$. Then $\mathcal{M}' \models O \odot_1 K(a) \cup \{\neg\beta\}$ and $\mathcal{M}' \models O \otimes_1 K(a) \cup \{\neg\beta\}$ because per definition $O \odot_1 K(a) = O \cup \{K \sqsubseteq K', K' \sqsubseteq K \sqcup \mathsf{msc}_O(a), K'(a)\}$ and $O \otimes_1 K(a) = O \cup \{K \sqsubseteq K', K'(a)\}$ and:

- $\mathcal{M}' \models O \cup \{\neg\beta\}$, because $K' \notin \mathcal{L}$ and $\mathcal{M}'$ is the same as $\mathcal{M}$ for all symbols in $\mathcal{L}$, and $\mathcal{M} \models O \cup \{\neg\beta\}$;
- $\mathcal{M}' \models (K \sqsubseteq K') \wedge (K' \sqsubseteq K \sqcup \mathsf{msc}_O(a)) \wedge (\mathcal{M}' \models K'(a))$ because of the construction of $\mathcal{M}'$.

The proof for negative literals $\alpha = \neg K(a)$ is done similarly by constructing a new model $\mathcal{M}'$ from a model $\mathcal{M} \models O \cup \{\neg\beta\}$ setting $\mathcal{M}'(K') = \mathcal{M}(K) \setminus \{\mathcal{M}(a)\}$.

Proof of Prop. 2 (p. 7). The proofs for the assertions in which $\circ_2$ is mentioned, i.e. (11), (14), (15), (18), will be done by proving it either for $\odot_2$ or for $\otimes_2$. The proof for the other operator then follows as a corollary using Obs. 1.

In the proofs the substitution $\sigma = [K/L, K'/K]$ will be used. Because of the fact that $O \subseteq (O \odot_2 K(a))_\sigma$ (see Obs. 1.6), the transformations of the models constructed in the proofs will be more readable. We will systematically use the fact that for all formulas F that do not contain L, F has a satisfying model iff F_σ has one.

Proof of (11): First assume $O \cup \{a \neq c\} \models K(c)$. Then also $O \circ_2 K(a) \cup \{a \neq c\} \models K'(c)$ and since $K' \sqsubseteq K \in O \circ_2 K(a)$ also $O \circ_2 K(a) \cup \{a \neq c\} \models K(c)$. Now let $\mathcal{M}$ be a model of $O \circ_2 K(a)$. If $\mathcal{M}(a) \neq \mathcal{M}(c)$, then $\mathcal{M} \models a \neq c$, and $\mathcal{M} \models K(c)$ follows. If, on the other hand, $\mathcal{M}(a) = \mathcal{M}(c)$, then because of $K(a) \in O \circ_2 K(a)$ also $\mathcal{M}(c) \in \mathcal{M}(K)$ results, i.e., $\mathcal{M} \models K(c)$.

Now assume $O \cup \{a \neq c\} \not\models K(c)$. Let $\mathcal{M}$ be a model of $O \cup \{a \neq c, \neg K(c)\}$. Consequently $\mathcal{M}(a) \neq M(c)$ and $\mathcal{M}(c) \notin \mathcal{M}(K)$. We have to show $O \odot_2 K(a) \not\models K(c)$. Applying the substitution σ to both sides of the entailment results in the task to show

$$O \cup \{L(a), K \sqsubseteq L, L \sqsubseteq K \sqcup \mathsf{msc_O}(a)\} \not\models L(c) \tag{19}$$

Construct a new model $\mathcal{M}'$ over $\mathcal{L}'' = \mathcal{L} \cup \{L\}$ from $\mathcal{M}$ as follows: $\mathsf{dom}(\mathcal{M}') = \mathsf{dom}(\mathcal{M})$, $\mathcal{M}'(S) = \mathcal{M}(S)$ for all symbols $S \in \mathcal{L}$ and $\mathcal{M}'(L) = \mathcal{M}(K) \cup \{\mathcal{M}(a)\}$. Then $\mathcal{M}'$ is a model of $O \cup \{\neg K(c)\}$ and additionally a model of $\{L(a), K \sqsubseteq K, L \sqsubseteq K \sqcup \mathsf{msc_O}(a), \neg L(c)\}$ showing (19). Applying Obs. 1.3 results in $O \otimes_2 K(a) \not\models K(c)$.

Proof of (12): First assume $O \models \neg K(c)$ and $O \models \neg\mathsf{msc_O}(a)(c)$. Then $O \odot_2 K(a) \models \neg K'(c)$ and because of $((K \sqcap \neg\mathsf{msc_O}(a)) \sqsubseteq K') \in O \odot_2 K(a)$ also $O \odot_2 K(a) \models (\neg K \sqcup \mathsf{msc_O}(a))(c)$ so that $O \odot_2 K(a) \models \neg K(c)$.

Now we want to show, if $O \not\models \neg K(c)$, then $O \odot_2 K(a) \not\models \neg K(c)$ and if $O \not\models \neg\mathsf{msc_O}(a)(c)$, then $O \odot_2 K(a) \not\models \neg K(c)$.
Assume $O \not\models \neg K(c)$. Let $\mathcal{M}$ be a model of $O \cup \{K(c)\}$ and construct $\mathcal{M}'$ as an extension of $\mathcal{M}$ with $\mathcal{M}'(L) = \mathcal{M}(K) \cup \{\mathcal{M}(c)\}$. Then $\mathcal{M}'(c) \in \mathcal{M}'(L)$ and $\mathcal{M}' \models (O \odot_2 K(a))_\sigma$ and so also $\mathcal{M}' \models (O \odot_2 K(a) \cup \{K(c)\})_\sigma$ resulting in $O \odot_2 K(a) \not\models \neg K(c)$.
Assume $O \not\models \neg\mathsf{msc_O}(a)(c)$. Let $\mathcal{M}$ be a model of $O \cup \{\mathsf{msc_O}(a)(c)\}$. Construct $\mathcal{M}'$ as an extension of $\mathcal{M}$ by setting $\mathcal{M}'(L) = \mathcal{M}(K) \cup \{\mathcal{M}(a), \mathcal{M}(c)\}$. Then as above $\mathcal{M}' \models (O \odot_2 K(a) \cup \{K(c)\})_\sigma$ and $O \odot_2 K(a) \not\models \neg K(c)$ results.

Proof of (13): Let $\mathcal{M} \models O \otimes_2 K(a)$; then the new model $\mathcal{M}'$ defined by $\mathsf{dom}(\mathcal{M}') = \mathsf{dom}(\mathcal{M})$, $\mathcal{M}'(S) = \mathcal{M}(S)$ for all symbols S different from K and $\mathcal{M}'(K) = \mathsf{dom}(\mathcal{M})$ is a model of $O \otimes_2 K(a)$ and of $K(c)$. (Remember that $K' \sqsubseteq K$ and $K(a)$ are the only formula of $O \otimes_2 K(a)$ that involve K.)

Proof of (14): As $K, K' \notin \beta$ we have $\beta\sigma = \beta$. First assume $O \models \beta$. We have to show $O \circ_2 K(a) \models \beta$. Applying σ this reduces to showing $(O \circ_2 K(a))_\sigma \models \beta$. But this is the case because of $O \subseteq (O \circ_2 K(a))_\sigma$ and the monotonicity of $\models$.

Now assume $O \circ_2 K(a) \models \beta$ for $\odot_2$ in place of $\circ_2$, i.e., applying σ again suppose that the following entailment holds:

$$O \cup \{L(a), K \sqsubseteq L, L \sqsubseteq K \sqcup \mathsf{msc_O}(a)\} \models \beta \tag{20}$$

Let $\mathcal{M}$ be a model over $\mathcal{L}(O \cup \{\beta\})$ of O. Extend $\mathcal{M}$ to $\mathcal{M}'$ by setting $\mathcal{M}'(L) = \mathcal{M}(K) \cup \{\mathcal{M}(a)\}$. Then $\mathcal{M}' \models O \cup \{L(a), K \sqsubseteq L, L \sqsubseteq K \sqcup \mathsf{msc_O}(a)\}$ and hence $\mathcal{M}' \models \beta$. As $\mathcal{M}$ is the reduct of $\mathcal{M}'$ to $\mathcal{L}(O \cup \{\beta\})$ also $\mathcal{M} \models \beta$. We have shown the assertion that if $O \odot_2 K(a) \models \beta$, then $O \models \beta$. The assertion for $\otimes_2$ in place of $\odot_2$ follows with Obs. 1.3.

The proofs of (15), (16) and (18) are similar. For (15) and (18) one constructs $\mathcal{M}'$ from a model $\mathcal{M} \models O \cup \{a \neq c, K(c)\}$ by setting $\mathcal{M}'(L) = \mathcal{M}(K) \setminus \{\mathcal{M}(a)\}$. For the proof of (16) one constructs the extension $\mathcal{M}'_1$ of $\mathcal{M}_1 \models O \cup \{K(c)\}$ by setting $\mathcal{M}'_1(L) = \mathcal{M}_1(K) \setminus \{\mathcal{M}_1(a)\}$. And one constructs the extension $\mathcal{M}'_2$ of $\mathcal{M}_2 \models O \cup \{\mathsf{msc_O}(a)(c)\}$ by setting $\mathcal{M}'_2(L) = \mathcal{M}_2(K) \setminus \{\mathcal{M}_2(a), \mathcal{M}_2(c)\}$

Proof of (17): Let $\mathcal{M} \models O \otimes_2 \neg K(a)$; then the new model $\mathcal{M}'$ defined by $\mathsf{dom}(\mathcal{M}') = \mathsf{dom}(\mathcal{M})$, $\mathcal{M}'(S) = \mathcal{M}(S)$ for all symbols S different from K and $\mathcal{M}'(K) = \emptyset$ is a model of $O \otimes_2 K(a)$ and of $\neg K(c)$.

References

1. Noy, N.F.: Semantic integration: A survey of ontology-based approaches. SIGMOD Record **33**(4) (2004) 65–70
2. Grove, A.: Two modellings for theory change. Journal of Philosophical Logic **17** (1988) 157–170
3. Wassermann, R., Fermé, E.: A note on prototype revision (1999) *"Spinning Ideas". (Electronic Essays dedicated to Peter Gärdenfors on his 50th Birthday).* http://www.lucs.lu.se/spinning/.
4. Özçep, Ö.L.: Ontology revision through concept contraction. In Artemov, S., Parikh, R., eds.: Proceedings of the Workshop on Rationality and Knowledge, 18th European Summerschool in Logic, Language, and Information, Universidad de Malaga, 7–11 August. (2006) 79–90
5. Küsters, R., Molitor, R.: Computing most specific concepts in description logics with existential restrictions. LTCS-Report 00-05, LuFG Theoretical Computer Science, RWTH Aachen, Germany (2000)
6. Alchourrón, C., Gaerdenfors, P., Makinson, D.: On the logic of theory change: partial meet contraction and revision functions. Journal of Symbolic Logic **50** (1985) 510–530
7. Katsuno, H., Mendelzon, A.: On the difference between updating a knowledge base and revising it. In Allen, J.F., Fikes, R., Sandewall, E., eds.: KR'91: Principles of Knowledge Representation and Reasoning. Morgan Kaufmann, San Mateo, California (1991) 387–394
8. Monk, D.J.: Mathematical Logic. Springer (1976)
9. Antoniou, G., Kehagias, A.: A note on the refinement of ontologies. International Journal of Intelligent Systems **15**(7) (2000) 623–632
10. Benferhat, S., Kaci, S., Berre, D.L., Williams, M.A.: Weakening conflicting information for iterated revision and knowledge integration. Artificial Intelligence **153**(1–2) (2004) 339–371
11. Meyer, T., Lee, K., Booth, R.: Knowledge integration for description logics. In Veloso, M.M., Kambhampati, S., eds.: AAAI, AAAI Press / The MIT Press (2005) 645–650
12. Qi, G., Liu, W., Bell, D.: A revision-based approach to handling inconsistency in description logics. In: Proceedings of the 11th International workshop on Non-Monotonic Reasoning (NMR06). (2006)
13. Gärdenfors, P., Rott, H.: Belief revision. In Gabbay, D., Hoger, C., Robinson, J., eds.: Handbook of Logic in Artificial Intelligence and Logic Programming. Volume 4. Oxford University Press (1995) 35–132