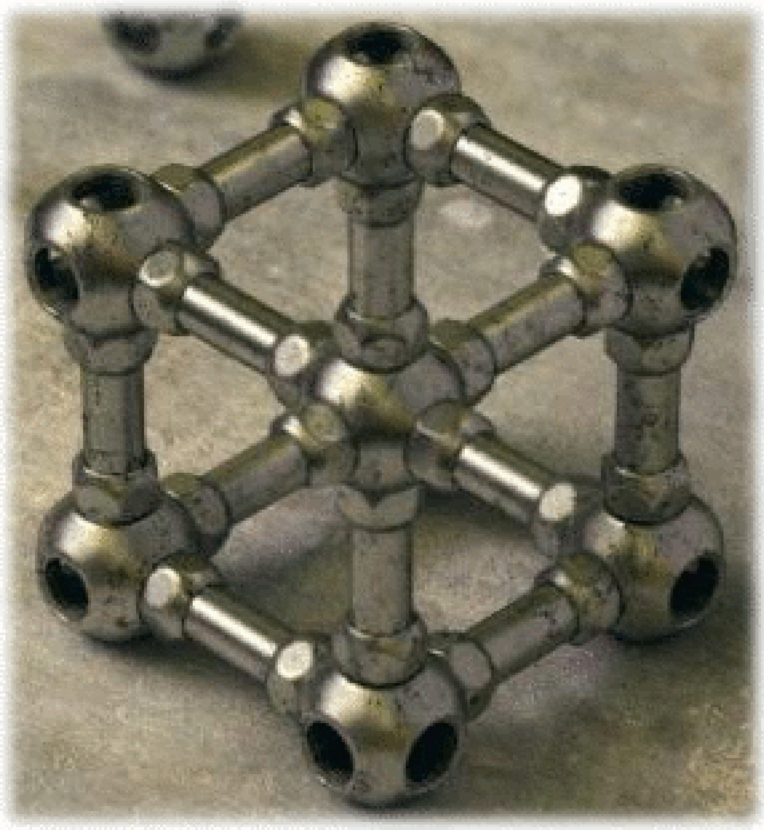

Base Revision for Ontology Debugging [1]

Authors:

Márcio Moretto Ribeiro

Renata Wassermann

[1] This work was supported by Fapesp Project LogProb, grant 2008/03995-5, São Paulo, Brazil.

Base Revision for Ontology Debugging

Márcio M. Ribeiro and Renata Wassermann
{marciomr,renata}@ime.usp.br
Department of Computer Science
Institute of Mathematics and Statistics
University of São Paulo, Brazil

Abstract

Belief Revision deals with the problem of adding new information to a knowledge base in a consistent way. Ontology Debugging, on the other hand, aims to find the axioms in a terminological knowledge base which caused the base to become inconsistent. In this paper, we propose a belief revision approach in order to find and repair inconsistencies in ontologies represented in some description logic. As the usual belief revision operators cannot be directly applied to description logics, we propose new operators that can be used with more general logics and show that, in particular, they can be applied to the logics underlying OWL-DL and Lite.

1 Introduction

With the advent of the Semantic Web [BLHL01], much attention has been devoted to the issue of representing terminological knowledge. Several languages for representing ontologies have been proposed and since 2004, the World Wide Web Consortium (W3C)[1] recommends *OWL*, in its three versions OWL-Full, OWL-DL and OWL-Lite [MvH04], as the standard language to represent ontologies on the web. Since OWL-Lite and OWL-DL are based on the description logics $\mathcal{SHIF}(\mathcal{D})$ and $\mathcal{SHOIN}(\mathcal{D})$ [HPS04] respectively, different reasoners for these languages were proposed that provide inference services to users. Standard services include verifying whether an ontology is consistent, whether a concept is a specialization of another one (classification) and finding the concepts of which an individual is an instance. These services are intended for static ontologies, i.e., they need to be combined with other techniques if we want to use them to deal with knowledge in the web, which is constantly evolving.

One approach to deal with the dynamics of a knowledge base is *Belief Revision* [Gär88, Han99b]. The idea of Belief Revision is to define operations to deal with the accommodation of new information. These operations are usually studied from two sides: rationality postulates establish the properties such

[1] http://www.w3.org/

operations should satisfy, while mathematical constructions define how the operations can be built. If we can prove that a construction and a set of rationality postulates are equivalent (a *representation theorem*), then we can concentrate on one side or on the other.

Since ontologies are not static, the standard reasoning services are not enough for the users. Dynamic ontologies change in time and may become inconsistent. As an example (adapted from [NM01]), let us consider an ontology that describes the domain of wines. Suppose that we have defined a concept Wine and three subconcepts Red Wine, White Wine and Rose Wine. For many years, all Zinfandel wines were red. Therefore, when we add a class to represent the Zinfandel wines we add it as a subclass of the Red Wine class. However, at a certain point wine makers begin to press the grapes and to take away the color-producing aspects of the grapes immediately, producing "White Zinfandel" whose color is actually rose. If we add an instance of White Zinfandel, with its property of having color rose to the class of Zinfandel, we would get an inconsistency, since all Zinfandels were said to be red. For this reason debugging services have been proposed [SC03, KPSH05]. Debugging services help the user to find which axioms are responsible for the inconsistency in the ontology.

The goal of this paper is to show how these debugging services can be linked to the belief revision field. That way, on the one hand, one can see debugging services through belief revision eyes. On the other hand, debugging services provide the means to implement belief revision techniques in description logics.

Section 2 introduces description logics, defines the syntax and the semantics of $\mathcal{SHOIN}(\mathcal{D})$ and $\mathcal{SHIF}(\mathcal{D})$ which are the logics that we are most interested in. Moreover, this section will show some properties that these description logics satisfy and some issues about negation of sentences in description logics. Section 3 discusses some rationality postulates for two belief revision operations: contraction and revision. In this section we argue about the importance of each of the postulates. Section 4 presents some constructions for these operations. In these section it is argued that the contraction operations presented in [HW02] can be applied on description logics, while the usual constructions for revision can not. Hence we present new constructions for revision that can be applied on description logics. Section 5 explains how to use algorithms for debugging ontologies to construct operations for belief revision. In the appendix we prove representation theorems linking sets of rationality postulates showed in section 3 and the constructions presented in section 4.

2 Description Logics

In this section we briefly introduce description logics. For more detail we refer to [BCM+03]. Description logics (DL) are a family of logics used to represent terminological knowledge. DLs have some advantages in comparison with earlier approaches like semantic networks, frame systems and first order logic. Differently from semantic networks and frame systems, DLs have formal semantics, and can be easily translated into a subset of first-order logics. Differently from first-order logic, inference in DLs is usually decidable.

The W3C standard language for representing ontologies on the web, OWL [DSB$^+$04], comes in three flavors: OWL-Lite, OWL-DL and OWL-Full. The latter is undecidable, as it allows, for example, the unrestricted use of transitive roles, which was proven in [HST99] to be undecidable. OWL-Lite was shown to be equivalent to the DL $\mathcal{SHIF}(\mathcal{D})$ while OWL-DL is equivalent to $\mathcal{SHOIN}(\mathcal{D})$ [HPS03]. For this reason these two DLs are of special interest for our work and will be used to exemplify the semantics of a description logics.

A Knowledge Base in description logics is a combination of two distinct sets: a TBox for terminological knowledge and an ABox for assertional knowledge. Starting from a set of atomic concepts and roles, new concepts and roles can be formed using the constructors in the language. The set of constructors and axiom types allowed in a particular description logic is what distinguishes a particular DL from the others.

The logic $\mathcal{ALC}$, for example, for concepts C and D and role R, allows for union ($C \sqcup D$), intersection ($C \sqcap D$), complement ($\neg C$) and value restrictions ($\forall R.C$ and $\exists R.C$). Besides the constructors, $\mathcal{ALC}$ allows concept subsumption ($C \sqsubseteq D$), concept equivalence ($C \equiv D$), concept assertions ($C(x)$) and role assertions ($R(x, y)$) as axioms. As an example, we can construct the concept of non-flying things as $\neg Fly$, the concept of the things which have wings as $\exists hasPart.Wing$ and we can write that penguins do not fly as $Penguins \sqsubseteq \neg Fly$, that Birds are the animals which have wings as $Birds \equiv Animal \sqcap \exists hasPart.Wing$, and that Tweety is a bird as $Bird(tweety)$.

The logic $\mathcal{SHIF}(\mathcal{D})$ allows all the constructors of $\mathcal{ALC}$ plus role hierarchy ($R \sqsubseteq S$), transitive roles, inverse roles, data-types and functional roles. For example in $\mathcal{SHIF}(\mathcal{D})$ we can write that the role $hasPart$ is the inverse of the role $isPartOf$. The logic $\mathcal{SHOIN}(\mathcal{D})$ allows all constructors of $\mathcal{SHIF}(\mathcal{D})$ plus nominals ($\{o_1, o_2, ...\}$) and cardinality restriction on roles ($\leq_n R$ and $\geq_n R$). Using cardinality restriction we can define something that has exactly two legs as $\leq_2 hasLeg \sqcap \geq_2 hasLeg$. For a complete definition of concepts and sentences in $\mathcal{SHOIN}(\mathcal{D})$ and $\mathcal{SHIF}(\mathcal{D})$ see [HPS03].

An interpretation is defined as a domain set Δ and an interpretation function $\mathcal{I}$. The interpretation function maps each atomic concept A to a subset $A^{\mathcal{I}}$ of Δ, each role $R^{\mathcal{I}}$ to a binary relation on Δ and each instance $o^{\mathcal{I}}$ to an element of Δ. This function then is extended to the complex concepts. We say that an interpretation satisfies an axiom if the axiom is valid in this interpretation.

We say that a sentence α is a semantic consequence of a set of sentences K if and only if every interpretation that satisfies K satisfies α and we write $K \models \alpha$. For example, if our TBox contains the sentences $Bird \sqsubseteq Fly$ and $Penguim \sqsubseteq Bird$ then the sentence $Penguim \sqsubseteq Fly$ can be inferred, because it is valid in every valid interpretation of the TBox. The consequence operator will be defined as $Cn(K) = \{\beta : K \models \beta\}$.

In the rest of the paper we will only consider that the logic has a language L and a consequence operator Cn. Every result will explicitly tell which properties the consequence operator must satisfy. Some of the properties a consequence operator can satisfy are summed up below:

Monotonicity: $B' \subseteq B \Rightarrow Cn(B') \subseteq Cn(B)$

Idempotency: $Cn(Cn(B)) = Cn(B)$

Inclusion: $B \subseteq Cn(B)$

Compactness: If $\alpha \in Cn(B)$ then there is $B' \subseteq B$ finite such that $\alpha \in Cn(B')$

Tarskian: Cn is tarskian iff it satisfies monotonicity, idempotency and inclusion.

2.1 Negation and Inconsistency

One characteristic of DLs that will be important in this work is that not every DL is closed under negation of sentences. For example in $\mathcal{SHIF}(\mathcal{D})$ and $\mathcal{SHOIN}(\mathcal{D})$ we have defined concept subsumption $\sqsubseteq$, but we have not defined the negation of a concept subsumption $\not\sqsubseteq$. In fact this is an open problem in description logics.

In [FHP$^+$06] the authors proposed a general definition of negation of sentences in DL. This definition can be applied for any DL, but negation may be hard to compute and the negation of a sentence does not need to be unique.

As we are trying to define a very general framework, it is important to define precisely what will be called inconsistency. One possible definition of inconsistent base is a base that implies everything, a trivial base. For DLs this means that a base is inconsistent if and only if it implies $\{\top \sqsubseteq \bot\}$. For example, suppose that we have the following belief base:

$$Bird \sqsubseteq Fly$$
$$Bird(tweety)$$
$$\neg Fly(tweety)$$

There is no interpretation that satisfies this base. In this case, any sentence can be inferred from the base as the contradiction trivializes it.

However, there are other definitions for inconsistency. For DLs it is argued that if a base implies $A \subseteq \emptyset$ for any concept A explicitly mentioned in the TBox than the base is inconsistent. For example:

$$\begin{array}{rcl} Bird & \sqsubseteq & Fly \\ Penguim & \sqsubseteq & Bird \\ Penguim & \sqsubseteq & \neg Fly \end{array}$$

This base is not trivial. The strange thing about this base is that *Penguim* can be inferred to be empty. Normally this means that a modelling error was committed and this is sometimes considered an inconsistency.

In this paper we will only assume that there is a set Ω of unwanted sentences. The only restriction for Ω is that $Cn(\emptyset) \cap \Omega = \emptyset$. An inconsistent belief base B is one that implies any of these sentences ($Cn(B) \cap \Omega \neq \emptyset$) and a sentence α is called inconsistent iff $Cn(\{\alpha\}) \cap \Omega \neq \emptyset$. For example, we could define $\Omega = \{\top \sqsubseteq \bot\}$ and use the first definition, we can define $\Omega = \{A \sqsubseteq \bot : A$ is an explicitly defined concept $\}$ and use the second definition or we can define Ω as any other unwanted set of sentences.

3 Belief Revision

Belief Revision [Gär88, GR95, Han99b] deals with the problem of accommodating new information in knowledge bases. In this work we use belief revision to study ontology dynamics. Most of the literature on belief revision has as a basis the AGM paradigm, that inherited its name from the initials of the authors of the seminal paper [AGM85]. Traditionally, three main operations are defined: contraction ($-$), expansion ($+$) and revision ($*$). These operations involve a knowledge base and an input sentence. Contraction is used when the agent wants to remove some information, expansion when it wants to add new information and revision when it wants to add information consistently into the knowledge base.

The AGM paradigm assumes that the belief of an agent are represented by a logically closed set of sentences, a *belief set*. From the three operations, only expansion can be uniquely defined: $K + \alpha = Cn(K \cup \{\alpha\})$. Revision and contraction are defined through rationality postulates, that state the properties any operation of revision or contraction should satisfy. In [AGM85], a construction for contraction and revision operations was proposed (*partial-meet operations*) and representation theorems were proven stating that any partial-meet contraction/revision satisfies the postulates for contraction/revision and any operation that satisfies the postulates can be constructed using partial-meet.

AGM theory is not restricted to classical propositional logic, but the consequence operator Cn is assumed to be tarskian, compact, satisfy the deduction theorem and supraclassicality. Following [FPA05a], we will refer to these properties as the AGM-assumptions. The AGM-assumptions exclude many interesting logics, such as many description logics.

Recently, it was shown that the AGM paradigm can not be applied to a broad class of description logics [FPA04, FPA05b]. It was shown that, in particular, there is no contraction (revision) in $\mathcal{SHIF}(\mathcal{D})$ or $\mathcal{SHOIN}(\mathcal{D})$ satisfying the AGM postulates. Although there are works [FPA06, RW06] proposing alternative sets of postulates that could be applied to these logics, these works, like the AGM paradigm, deal with belief sets. Belief sets are usually infinite and not very practical from the computational point of view. This led some authors [Neb90, Han91] to consider revision in *belief bases*, i.e., knowledge bases which are not necessarily closed.

In the belief base approach the expansion is defined as $B + \alpha = B \cup \{\alpha\}$, the operations of revision and contraction are also defined in terms of rationality postulates and constructions, linked by representation theorems. We can concentrate our analysis in this section on the rationality postulates, without worrying about the details of the constructions. The representation theorems assure that these properties completely describe the operations.

It is important to notice that the belief base and the belief set approaches are very different. The belief set approach does not distinguish between explicit and inferred knowledge. On the one hand, that is much closer to the knowledge level [New82] ideal. On the other hand, in the belief set approach an agent must consider many irrelevant sentences every time it performs a revision or a contraction. For example, if an agent explicitly believes that $Penguim(tweety)$, it must also belief that $(Penguim \sqcup Blue)(tweety)$. In the belief set approach the reason for believing these two sentences are indistinguishable, so if the first one is removed from the belief set the agent still have to choose whether to retain the second sentence or not. In the belief base approach, the second sentence is treated merely as a consequence of the belief base, so if the first sentence is removed from the belief base the second automatically disappears from the consequences.

In this work we will follow the belief base approach, because we are interested in the link between belief base theory and implementation. Moreover, we will show that the operations for belief bases can be easily implemented using algorithms already studied in the ontology debugging literature [SC03, KPSH05].

In the next sections we will show some rationality postulates and comment on their adequacy. Then we will show some constructions and the respective representation theorems. In what follows, K will always stand for a belief set and B for a belief base.

3.1 Postulates for contraction

An agent performs a contraction if it wants to remove some information α from its belief base B. It is usually assumed in the literature that the output of a contraction operation should be a subset of the original belief base that does not imply the input sentence. This assumption can be captured by two rationality postulates: success and inclusion.

(success) If $\alpha \notin Cn(\emptyset)$ then $\alpha \notin Cn(B - \alpha)$

Satisfying success means that after the contraction is performed, the agent should not believe the input, i.e., the contracted belief base should not imply the input sentence. The only exception is when α is a tautology, since it will be implied by any set of formulas.

(inclusion) $B - \alpha \subseteq B$

Satisfying inclusion means that no new sentence should be added when performing a contraction.

A contraction operation that removes all sentences from the belief base satisfies inclusion and success. However, we usually want to remove α changing the original belief base as little as possible. This is known as the "Principle of Minimal Change".

In the AGM paradigm the postulate that plays this role is recovery:

(recovery) $K - \alpha + \alpha = K$

Recovery states that the result of contracting a belief set by α and then expanding the resulting set by α should have as output the original belief set. The idea is that enough sentences should be retained in the contraction so that the original belief set can be recovered.

However, it has been argued in the literature that recovery is not a good postulate for belief bases [Fuh91, Flo06].

Other postulates for minimality were proposed by Hansson [Han99b]. These postulates, core-retainment and relevance, state intuitively that a sentence can only be removed from a belief base if it is relevant, in some sense, to infer the input.

(core-retainment) If $\beta \in B$ and $\beta \notin B - \alpha$, then there is B' such $B' \subseteq B$ and $B' \nvdash \alpha$, but $B' \cup \{\beta\} \vdash \alpha$.

(relevance) If $\beta \in B$ and $\beta \notin B - \alpha$, then there is B' such $B - \alpha \subseteq B' \subseteq B$ and $B' \nvdash \alpha$, but $B' \cup \{\beta\} \vdash \alpha$.

Despite the similarity of these postulates, core-retainment is more general. Every contraction satisfying relevance satisfies core-retainment, but the converse is not true. For example, take the following belief base:

$$Bird \sqsubseteq Fly$$
$$\exists hasPart.Wing \sqsubseteq Fly$$
$$Penguim \sqsubseteq \exists hasPart.Wing$$
$$Penguim \sqsubseteq Bird$$
$$Penguim(tweety)$$

Suppose now that we want to contract this base by $Fly(tweety)$ and that $Penguim(tweety)$ and $Penguim \sqsubseteq Bird$ were both removed. This contraction satisfies core-retainment, but it does not satisfy relevance.

The last aspect of contraction that we are going to mention is syntactic independence. The AGM paradigm states that contraction should treat equivalent sentence equally, through the extensionality postulate:

(extentionality) If $Cn(\alpha) = Cn(\beta)$ then $K - \alpha = K - \beta$

However, for belief bases this postulate allows some non-intuitive operations. Consider the following belief base B:

$$Bird(tweety)$$
$$Bird \sqsubseteq Fly$$

A contraction operation that makes $B - Fly(tweety) = \{Bird(tweety)\}$ and $B - (Bird \sqcup Penguim)(tweety) = \{Bird \sqsubseteq Fly\}$ satisfies extensionality. However, although $Bird(tweety)$ and $(Bird \sqcup Penguim)(tweety)$ are not equivalent, any subset of the belief base implies the first sentence if and only if it implies the second. Hence, it would not be reasonable that these two contractions had different results. The following postulate encompasses these cases:

(uniformity) If for all subsets B' of B it holds that $\alpha \in Cn(B')$ iff $\beta \in Cn(B')$ then $B - \alpha = B - \beta$.

We have seen in this section that a contraction should: remove the input from the consequences of the resulting base (success), add nothing new (inclusion), change as little as possible in this process (minimality) and treat equivalent sentences equally (uniformity).

In the next section we will show some postulates a revision operation should satisfy.

3.2 Postulates for revision

Revision is applied when an agent wants to add some sentence consistently to his knowledge base, i.e., we expect as output a consistent knowledge base containing the information that was added. Our first requirements for a revision operation are consistency and success:

(consistency) $Cn(B * \alpha) \cap \Omega = \emptyset$

(success) $\alpha \in B * \alpha$

Consistency means that no sentence in Ω should be implied by the resulting belief base. Success states that α should be in the resulting base.

That brings us a problem: What if the input implies some sentence in Ω? Then either success or consistency are satisfied, but not both. So if we want success to be satisfied then we need a weaker version of consistency, and vice-versa.

(weak-consistency) If $Cn(\alpha) \cap \Omega = \emptyset$ then $Cn(B * \alpha) \cap \Omega = \emptyset$

(weak-success) If $Cn(\alpha) \cap \Omega = \emptyset$ then $\alpha \in B * \alpha$

A revision can satisfy success together with weak-consistency or weak-success together with consistency. The second choice seems more intuitive, but the first one is closer to the AGM paradigm.

There are several works in the literature [Gal92, Mak97, Han97]that deal with operations similar to revision but that do not impose success. This type of operation is called non-prioritized revision [Han99a], since it does not assign priority to the input. One such operation is *semi-revision*, proposed by Hansson in [Han97].

An agent should perform a semi-revision when it is not sure about the new information and wants to postpone the decision of whether the sentence should be accepted or not. Semi-revision in DLs was studied in [HWKP06].

In order to guaranty the consistency of a belief base the agent sometimes needs to remove some sentences from the belief base. We should not add more than would be added in a simple expansion by the input. This restriction is captured by the inclusion postulate:

(inclusion) $B * \alpha \subseteq B + \alpha$

As in contraction, we want to change our belief base as little as possible. Postulates for minimality for revision are very similar to the ones presented for contraction. The same arguments about minimality postulates given for contraction can be given for revision.

(core-retainment) If $\beta \in B$ and $\beta \notin B * \alpha$, then there is B' such $B' \subseteq B \cup \{\alpha\}$ and $Cn(B') \cap \Omega = \emptyset$, but $Cn(B' \cup \{\beta\}) \cap \Omega \neq \emptyset$.

(relevance) If $\beta \in B$ and $\beta \notin B * \alpha$, then there is B' such $B * \alpha \subseteq B' \subseteq B \cup \{\alpha\}$ and $Cn(B') \cap \Omega = \emptyset$, but $Cn(B' \cup \{\beta\}) \cap \Omega \neq \emptyset$.

In this work we are going to use two other rationality postulates. The first one is a rationality postulate that states that every time a belief base is revised by any of its own elements then the resulting base should be the same. This postulate is normally associated with semi-revision and is called *internal exchange*

(internal exchange) If $\alpha, \beta \in B$ then $B * \alpha = B * \beta$

The last one states that if a belief base is expanded by a sentence α and then revised by α the result should be the same as just revising the original base by α:

(pre-expansion) $B + \alpha * \alpha = B * \alpha$

In the next section, we will present constructions for contraction and revision and the representation theorems connecting the constructions to the rationality postulates.

4 Constructions

In this section we will present some constructions for contraction and revision of belief bases. The constructions for contraction were already discussed in the literature [Han99b] and it was shown in [HW02] that they can be applied to any monotonic and compact logic. Many Description Logics, in particular $\mathcal{SHIF}(\mathcal{D})$ and $\mathcal{SHOIN}(\mathcal{D})$ are monotonic and compact. Constructions for base revision, on the other hand, usually assume that the logic is closed under negation of sentences. But many DLs are not, thus, the usual constructions cannot be applied. In this section we will present constructions for revision that do not depend on negation. These constructions are more general than the traditional ones and can also be applied to any monotonic and compact logic. For each construction proposed, a representation theorem links it with a set of postulates.

4.1 Contraction

In this section, we will present two different constructions for contraction of belief bases. The first one, *partial meet contraction*, is the belief base counterpart of the traditional AGM construction presented in [AM82]. The second one, *kernel contraction*, was proposed in [Han94] and is a generalization of AGM safe contraction [AM85].

Although these constructions seem very different, they share many properties. The representation theorem for these constructions show that the only difference between them, with respect to the rationality postulates, is the postulate for minimality. Partial meet contraction satisfies relevance while kernel contraction satisfies only core-retainment. Hence, kernel contraction is more general than partial meet.

4.1.1 Partial Meet Contraction

Partial meet contraction of a belief base B by a sentence α consists in selecting some maximal subsets of B that do not imply α and taking their intersection.

The set of maximal subsets of B that do not imply α is called *remainder set of B and α* and denoted by $B \perp \alpha$:

Definition 1 (Remainder Set) $B \perp \alpha = \{B' \subseteq B$ *such that* $\alpha \notin Cn(B')$ *and if* $B' \subset B''$ *then* $\alpha \in Cn(B'')\}$

The function that chooses at least one element of the remainder set is called *selection function*:

Definition 2 (Selection Function) *A function γ is a selection function if it satisfies:*

- $\emptyset \neq \gamma(B \perp \alpha) \subseteq B \perp \alpha$ *if* $B \perp \alpha \neq \emptyset$
- $\gamma(B \perp \alpha) = \{B\}$ *otherwise*

Partial meet contraction is then formally defined as the intersection of the subsets of B chosen by the selection function:

Definition 3 (Partial Meet Contraction) $B -_\gamma \alpha = \bigcap \gamma(B \perp \alpha)$

The representation theorem for partial meet contraction shows that this construction is equivalent to a set of rationality postulates:

Representation Theorem 1 (Partial Meet Contraction) *[Han92a] An operation $-$ satisfies success, inclusion, relevance and uniformity iff it is a partial meet contraction for some γ.*

In [HW02], it was shown that this representation theorem holds for any compact and monotonic logic.

Notice that the function γ is not fully determined. Each γ will return a different resulting belief base. The representation theorem proves that, in one hand, no matter which γ is chosen the resulting base will satisfy the set of postulates. On the other hand, any contraction that satisfies this set of postulates can be constructed as a partial meet contraction as long as the right γ is chosen.

4.1.2 Kernel Contraction

Another way to construct a contraction operator is removing some elements of each minimal subset of B that implies α. This kind of construction is called kernel contraction. The set of minimal subsets of B that imply α is called *kernel of B and α* and denoted by $B \perp\!\!\!\perp \alpha$:

Definition 4 (Kernel) $B \perp\!\!\!\perp \alpha = \{B' \subseteq B$ *such that* $\alpha \in Cn(B')$ *and if* $B'' \subset B'$ *then* $\alpha \notin Cn(B'')\}$

The function that chooses at least one element of each element of the kernel is called *incision function*:

Definition 5 (Incision Function) *A σ is an incision function if it satisfies:*

- $\sigma(B \perp\!\!\!\perp \alpha) \subseteq \bigcup(B \perp\!\!\!\perp \alpha)$
- *If* $\emptyset \neq X \in B \perp\!\!\!\perp \alpha$ *then* $X \cap \sigma(B \perp\!\!\!\perp \alpha) \neq \emptyset$

Kernel contraction is defined as the original belief base with the elements chosen by the incision function removed:

Definition 6 (Kernel Contraction) $B -_\sigma \alpha = B \setminus \sigma(B \perp\!\!\!\perp \alpha)$

Like partial meet contraction, kernel contraction depends on the choice of a function. Kernel contraction is equivalent to a set of rationality postulates for contraction:

Representation Theorem 2 (Kernel Contraction) *[Han94] The operator $-$ is a kernel contraction if and only if it satisfies the following postulates for contraction: success, inclusion, core-retainment and uniformity.*

The representation theorem for kernel contraction also holds for any compact and monotonic logic, as shown in [HW02]. In [HW02], there are also representation results for revision operators based on these two constructions for contraction. However, these constructions assume that the logic is closed by negation of sentences and the representation theorems depend on properties of negation. In the next section, we will present constructions for revision operators that do not depend on the existence of negation and show representation theorems that hold in any compact and monotonic logic.

4.2 Revision without negation

Following [Han99b] constructions for revision can be defined using the constructions for contraction already presented. An external revision is defined as $B+\alpha-\neg\alpha$ and an internal revision is defined as $B-\neg\alpha+\alpha$. In [HW02], representation theorems for internal/external partial meet and kernel revision were proven. However, these theorems depend on each sentence α having a negation $\neg\alpha$ that satisfies a property called α-local non-contravention.[2] We have already mentioned in section 2 that the definition of negation of a sentence in some Description Logics is still an open issue. In this section, we will present alternative constructions for revision that, like the constructions for contraction, can be used for any logic that is compact and monotonic.

We will also present constructions for semi-revision. As mentioned before, semi-revision is an operation of non-prioritized revision, i.e., a revision operation where the input does not always have the highest priority. All the constructions in this section follow a certain pattern: we first expand the belief base B by the input α, and then use a generalized version of partial meet or kernel contraction to contract by Ω. The choice between partial meet or kernel construction leads to two types of revision. Each of these types come in three flavors depending on the constrains imposed to the selection or incision function: semi-revision, revision with weak-success and revision with strong success. Hence, we will define six constructions here.

We will only work here with external revision [Han92b] for a simple reason. Although originally both internal and external revision were defined in terms of contraction by the negation of the input, in external revision this dependence on negation can be avoided by imposing conditions on the selection or incision functions, as will be seen. Internal revision is the belief base version of the traditional AGM revision. The main difference between internal and external belief base revision is in the way they deal with equivalent sentences. Internal revision satisfies uniformity, while external revision only satisfies a very weak version of this postulate. Our constructions for revision without negation do not even satisfy this weak form of uniformity. This is due to the idea of protecting the input, our choice mechanisms (selection or incision functions) strongly depend on the input sentence.

We have presented in the last section partial meet and kernel contractions involving a belief base and a sentence. The proposed generalization for inconsistency allows it to be a set of sentences. For this reason we need to generalize the constructions of partial meet and kernel contraction to accept sets of sentences as inputs. This generalization is straightforward.

The generalized remainder set of B and A, where both B and A are sets of sentences, is the set of maximal subsets of B that does not imply any element of A. Notice that the previous definition is a particular case of this one when A is a singleton.

Definition 7 (Generalized Remainder Set) $B \perp A = \{B' \subseteq B$ *such that*

[2] A logic satisfies α-local non-contravention iff, if $\neg\alpha \in Cn(B \cup \{\alpha\})$ then $\neg\alpha \in Cn(B)$

$A \cap Cn(B') = \emptyset$ *and if* $B' \subset B''$ *then* $A \cap Cn(B'') \neq \emptyset\}$

The generalized kernel is the set of minimal subsets of B that implies at least one element of A. The previous definition of kernel is a particular case of this one too.

Definition 8 (Generalized Kernel) $B \perp\!\!\!\perp A = \{B' \subseteq B$ *such that* $A \cap Cn(B') \neq \emptyset$ *and if* $B'' \subset B'$ *then* $A \cap Cn(B'') = \emptyset\}$

Generalized versions of partial meet and kernel contractions can be obtained by substituting the original definition of remainder and kernel sets by their generalized version.

In the rest of this section, we will show the six constructions and the corresponding representation theorems. All the theorems hold for any compact and monotonic logic.

4.2.1 Semi-revision

Semi-revision was proposed in [Han97] as an operation that may or not accept the input sentence, depending on the choices made. The idea is to expand the belief base by the input and then contract by Ω. If the input caused the expanded belief base to become inconsistent, the contraction by Ω may remove the input in order to restore consistency.

Here, since we want to consider inconsistency as a set, semi-revision is constructed by expanding the original belief base by the input α and then contracting by Ω using the generalized version of kernel or partial meet contraction.

Definition 9 (Semi-revision) *[Han97]* $B?\alpha = (B + \alpha) - \Omega$

Semi-revision satisfies inclusion, consistency, internal exchange. If the contraction is a kernel contraction then it satisfies core-retainment and if it is a partial meet contraction it satisfies relevance. All the proofs for the representation theorems can be found in appendix A.

Representation Theorem 3 (Kernel Semi-revision (KSR)) *The operation* $?$ *is a kernel semi-revision iff it satisfies: inclusion, consistency, core-retainment, pre-expansion and internal exchange.*

Representation Theorem 4 (Partial Meet Semi-revision (PMSR)) *The operation* $?$ *is a partial meet semi-revision iff it satisfies: inclusion, consistency, relevance, pre-expansion and internal exchange.*

The other constructions for revision that we will present were inspired in semi-revision. They all follow this pattern: first expand by α and then contract by Ω with partial meet or kernel contraction. But we add extra constraints to the selection or incision functions in order to assign higher priority to the input sentence.

4.2.2 Revision with weak success

A partial meet revision with weak success (PMWS) is based on a selection function that protects consistent inputs:

Definition 10 (Selection Function that Protects Consistent Inputs) *A selection function that protects consistent inputs is defined as:*

- $\emptyset \neq \gamma(B \perp \Omega, \alpha) \subseteq B \perp \Omega$
- *If* $Cn(\alpha) \cap \Omega = \emptyset$ *then* $\alpha \in \bigcap \gamma(B \perp \Omega, \alpha)$

This means that whenever the input is consistent, it will be part of all the selected remainders.

Representation Theorem 5 (Partial meet revision with weak-success) *The operator $*$ is a partial meet revision without negation with weak success if and only if it satisfies the following postulates for revision: weak-success, consistency, inclusion, relevance and pre-expansion.*

A kernel revision without negation with weak-success (KWS) uses an incision function that protects consistent inputs:

Definition 11 (Incision Function that Protects Consistent Inputs) *An incision function that protects consistent inputs is defined as a function σ that satisfies:*

- $\sigma(\alpha, B \perp\!\!\!\perp \Omega) \subseteq \bigcup(B \perp\!\!\!\perp \Omega)$
- *If* $\emptyset \neq X \in B \perp\!\!\!\perp \Omega$, *then* $X \cap \sigma(\alpha, B \perp\!\!\!\perp \Omega) \neq \emptyset$
- *If* $\Omega \cap Cn(\{\alpha\}) = \emptyset$, *then* $\alpha \notin \sigma(\alpha, B \perp\!\!\!\perp \Omega)$

This means that whenever the input sentence is consistent, it will be retained in the revised belief base.

Representation Theorem 6 (Kernel revision with weak-success) *The operator ? is a kernel revision without negation with weak success if and only if it satisfies the following postulates for revision: weak-success, consistency, inclusion, core-retainment and pre-expansion.*

Revision with weak success satisfies the success postulate only for consistent inputs. Success is sacrifices in order to obtain consistency.

4.2.3 Revision with full success

A partial meet revision without negation with success (PMS) is defined using a selection function that protects the input:

Definition 12 (Selection Function that Protects the Input) *A selection function that protects the input is defined as*

- $\emptyset \neq \gamma(B\perp\Omega, \alpha) \subseteq B\perp\Omega$ *and* $\alpha \in \bigcap \gamma(B\perp\Omega, \alpha)$ *if* $Cn(\{\alpha\}) \cap \Omega = \emptyset$
- $\gamma(B\perp\Omega, \alpha) = \{B\}$ *otherwise*

Representation Theorem 7 (Partial meet revision with success) *The operator $*$ is a partial meet revision with weak success if and only if it satisfies the following postulates for revision: success, weak-consistency, inclusion, relevance and pre-expansion.*

A kernel revision with success (KS) is constructed using a incision function that protects the input:

Definition 13 (Incision Function that Protects the Input) *An incision function that protects the input is defined as a function σ that satisfies:*

- $\sigma(\alpha, B \perp\!\!\!\perp \Omega) \subseteq \bigcup(B \perp\!\!\!\perp \Omega)$
- *If* $\Omega \cap Cn(\{\alpha\}) = \emptyset$ *and* $\emptyset \neq X \in B \perp\!\!\!\perp \Omega$, *then* $X \cap \sigma(\alpha, B \perp\!\!\!\perp \Omega) \neq \emptyset$
- $\alpha \notin \sigma(\alpha, B \perp\!\!\!\perp \Omega)$

The last condition assures that the input will never be removed.

Representation Theorem 8 (Kernel Revision with Success) *The operator $*$ is a kernel revision without negation if and only if it satisfies the following postulates for revision: success, weak-consistency, inclusion, core-retainment, pre-expansion.*

These two operations are in line with the AGM paradigm, in that they sacrifice consistency in order to have unconditional success.

4.2.4 Discussion

We have defined two constructions for semi-revision (kernel and partial meet) and four constructions for revision (kernel revision that protects the input, kernel revision that protects consistent inputs, partial meet revision that protects the input and partial meet revision that protects consistent inputs).

Each construction is equivalent to a set of postulates as proved by the representation theorems in appendix A. The constructions that use kernel contraction satisfy core-retainment, while the ones that use partial meet contraction satisfy relevance. The constructions that use selection (incision) function that protects the input satisfy success and weak-consistency, while the ones that use

Name	KSR	PMSR	KS	KWS	PMS	PMWS
Inclusion	yes	yes	yes	yes	yes	yes
Consistency	strong	strong	weak	strong	weak	strong
Success	no	no	strong	weak	strong	weak
Syntactic Independ.	internal exchange	internal exchange	no	no	no	no
Pre-Expansion	yes	yes	yes	yes	yes	yes
Minimality	core-ret.	relev.	core-ret.	core-ret.	relev.	relev.

selection (incision) function that protects consistent inputs satisfy weak-success and consistency. Every operation satisfy inclusion.

In fact, the representation theorems showed in appendix A are stronger than that. They prove that each of these constructions is fully characterized by a specific set of rationality postulates. Table 4.2.4 sums up the content of these theorems

5 Implementation

In the introduction we have claimed that the link between debugging services for DLs and formal properties provided by the belief revision approach has not been deeply studied yet. In this section we are going to show how to link these two areas.

One classical service provided by DL reasoners is consistency checking. In the last few years it has been noticed that telling that an ontology is inconsistent is not enough for the user [SC03]. Manually finding what caused the inconsistency and how it can be repaired can be very hard. The purpose of the debugging services developed recently [SC03, KPSH05, KPS05] is to guide the user in this process. In this section, we are going to show some algorithms for *axiom pinpointing*. Axiom pinpointing [SC03] consists in finding all the *justifications* of a sentence α with respect to a knowledge base B, which are the minimal subsets of B that imply α. In other words, axiom pinpointing consists in finding the kernel of a knowledge base B with respect to α.

We do not intend to present details of the implementation, we just want to show how this practical problem of ontology debugging can be linked to the theoretical approach of belief revision.

In [SC03] the authors presented algorithms for axiom pinpointing based on the idea of finding what they call "Minimal Unsatisfiability Preserving Sub-TBox" (MUPS) in the logic $\mathcal{ALC}$. This work was then extended to $\mathcal{SHIF}(\mathcal{D})$ and $\mathcal{SHOIN}(\mathcal{D})$ in [KPSH05, Kal06] and more recently generalized to several logics in [BP07].

Kernels (or MUPS) can be computed using "black-box" techniques, that call a reasoner as a subroutine that tells if a sentence is implied by the knowledge base, or using "glass-box" techniques, that modify existing inference mecha-

```
BLACKBOX(B, α)
B' ← ∅
for β ∈ B
do B' ← B' ∪ {β}
    if α ∈ Cn(B')
        then break
for ϵ ∈ B'
do if α ∈ Cn(B' \ {ϵ})
        then B' ← B' \ {ϵ}
return B'
```

Figure 1: Black Box algorithm

nisms. In [Kal06], two main techniques were proposed in order to find kernels, a black-box one and a hybrid solution that uses a glass-box step combined to the black-box approach.

The author presents a black-box algorithm for axiom pinpointing called "Expand-Shrink" which is presented in figure 5. This algorithm can be split in two parts: first expanding an initially empty knowledge base B' with axioms of the original base B and then pruning it. Notice that the pruning part is what guaranties the correctness of the algorithm. The pruning could be applied directly to the original knowledge base, although the size of the original base may turn this unfeasible.

DL reasoners based on tableaux decide whether an axiom is entailed by a knowledge base by trying to construct a model for the knowledge base together with the negation of the axiom. The idea of a glass box algorithm [KPS05] is to keep track of the axioms used to prove this entailment. This process is called tracing.

As argued before, the pruning part of the black-box algorithm could be applied directly to the original knowledge base if its size was not too prohibitive. The idea of hybrid techniques is to use the glass-box algorithm to shrink the knowledge base. This way it can be used as a first step in the black box algorithm in order to make it more efficient.

The hybrid approach can be used to find one element of the kernel. Once one element of the kernel is computed, the others can be found using the algorithm showed in figure 5. This is a recursive algorithm that, given one element of the kernel, returns the whole kernel.

Now, if we want find the generalization of the kernel that accepts two sets of sentences as input, we need to find the kernel of the belief base B with respect to each element of A and then remove each element that is not minimal. Only elements that are properly contained in another one have to be removed. The correctness of this algorithm is proved in appendix B.

To find the remainder set in order to use the partial meet contraction we could follow [MLBP06] that presents an algorithm to find maximal consistent

KERNEL(B, α)
if $\alpha \notin Cn(B)$
    then return
$min \leftarrow blackbox(B, \alpha)$
$B \perp\!\!\!\perp \alpha \leftarrow B \perp\!\!\!\perp \alpha \cup \{min\}$
for $\beta \in min$
do
      $B \perp\!\!\!\perp \alpha \leftarrow B \perp\!\!\!\perp \alpha \cup kernel(B \setminus \{\beta\}, \alpha)$
return

Figure 2: Algorithm to find the kernel of B w.r.t. α

subset of a TBox or we can extract it from the kernel using the Reiter's algorithm [Rei87] as showed in [Was00]. The algorithm consists in finding minimal incisions of the kernel. Each of this minimal incisions corresponds to one element of the remainder set and vice-versa.

6 Related Work

Recently, much attention has been devoted to the study of the dynamics of terminological knowledge. This interest is due to the development of languages to represent terminological knowledge in the web which is a dynamic environment. This issue has been studied from the theoretical point of view, showing that AGM operations can not be applied to many description logics [FPA05a] and from a practical point of view with the development of debugging services for DL reasoners [SC03, Kal06].

The dynamic of ontologies has been studied under the name ontology evolution. A good overview of it is found in [HS04]. We believe that the major contribution in trying to apply belief revision to ontologies is the separation between postulates and construction. The postulates provide a complete description of the construction. Therefore the operations of revision or contraction can be studied in an abstract level.

There are not many works in the literature that explored ontology evolution following the belief revision approach. In [FPA05a] the authors showed that AGM contraction cannot be applied to some important description logics like $\mathcal{SHIF}(\mathcal{D})$ and $\mathcal{SHOIN}(\mathcal{D})$. The main difference from our work is that they restrict themselves to classical AGM operations while we deal with belief bases. The problem they encountered with the recovery postulate is not present in operations on belief bases. We showed that Hansson's contractions apply to belief bases in $\mathcal{SHIF}(\mathcal{D})$ and $\mathcal{SHOIN}(\mathcal{D})$, but that there was a problem concerning revision.

[HWKP06] studied the operation of semi-revision for belief bases in $\mathcal{SHIF}(\mathcal{D})$ and $\mathcal{SHOIN}(\mathcal{D})$ as we mentioned earlier in this paper. Our main contribution to with respect to this work was to present sets of postulates for revision of belief bases that include success (or weak success) and still can be applied to $\mathcal{SHIF}(\mathcal{D})$ and $\mathcal{SHOIN}(\mathcal{D})$.

In [KL07], there is a proposal of applying belief revision to ontologies using a construction based on epistemic entrenchment. The authors suggest an implementation, but there is no proof of formal properties of the operations.

7 Conclusion

Although AGM operations cannot be applied to many DLs that does not mean that we should discard using belief revision theory to study ontology dynamics. Belief revision theory has an important characteristic: it does not only describe constructions, but tries to provide rationality postulates that fully characterize the operations of knowledge dynamics. Instead of using the traditional AGM postulates, we explored belief base postulates. We showed some rationality postulates for contraction and revision. Then we showed how to construct these operations. Construction for contraction presented in [HW02] can be applied to any compact and monotonic logics, thus, differently from the AGM paradigm they can be applied to most DLs including $\mathcal{SHIF}(\mathcal{D})$ and $\mathcal{SHOIN}(\mathcal{D})$. However, the traditional constructions for revision cannot be applied, because they depend on the definition of negation of sentences. So we proposed new constructions for revision that can be applied to any monotonic and compact logic.

We propose two constructions for semi-revision, partial-meet and kernel, and four constructions for revision: partial-meet with weak-success, partial-meet with strong-success, kernel with weak-success and kernel with strong-success. Since inconsistencies in DLs can be defined in many ways and we wanted our work to be as general as possible we have generalized the definition of inconsistency. We proved a representation theorem for each of these constructions using this generalized version of inconsistency. The representation theorems show which set of rationality postulates is equivalent to each construction. Ontology debugging algorithms provide means to implement these constructions.

That way we have linked the areas of ontology debugging and belief revision. This is important since belief revision can provide theoretical background to ontology debugging while algorithms for debugging can be used to implement belief revision constructions.

Future work includes implementing and testing each of the operations described here. Another interesting issue for future work is weakening the inclusion postulate for contraction.

At a first look this postulate seems very natural. So natural that this postulate together with success are sometimes considered the minimum requirement for a contraction. However that is not always what is expected. For example assume that we have the following belief base:

$$Bird \sqsubseteq Fly$$
$$Bird(tweety)$$

Suppose that we want to contract the sentence $Fly(tweety)$. If the contraction satisfies success and inclusion we have three choices: remove $Bird \sqsubseteq Fly$, remove $Bird(tweety)$ or remove both. However, sometimes these options are too strong. Maybe we just want to consider *tweety* as an exception. There are cases in which we do not want to remove a sentence, but just change it. For example we could change the sentence $Bird \sqsubseteq Fly$ to $Bird \sqsubseteq Fly \sqcup \{tweety\}$, and treat *tweety* as an exception. This is forbidden by the inclusion postulate because the sentence $Bird \sqsubseteq Fly \sqcup \{tweety\}$ has to be added to the resulting belief base. A weaker version of the inclusion postulate that only requires that$B - \alpha \subseteq Cn(B)$ could be considered.

Acknowledgements: The first author is supported by FAPESP and the second is partially supported by CNPq. This work is part of the project "Logical Consequence and Combinations of Logics" funded by FAPESP, grant number 2004/14107-2.

Appendix

A Representation Theorems

In order to prove the representation theorems we are going to use the following lemmas:

Lemma 1 *[AM81] $B \perp A = \emptyset$ if and only if $Cn(\emptyset) \cap A \neq \emptyset$*

Lemma 2 (Upper Bound Property) *[AM81] If $X \subseteq B$ and $Cn(X) \cap A = \emptyset$ then there is X' such that $X \subseteq X' \in B \perp A$*

Lemma 3 (Inconsistent Expansion) *[Han99b] If $K * a$ satisfies relevance and success then it satisfies* inconsistent expansion*:*
if $\Omega \cap Cn(\{\alpha\}) \neq \emptyset$ then $K * \alpha = K + \alpha$.

Representation Theorem 3 [Kernel Semi-revision](adapted from [Han97])
The operation $?_\sigma$ is a kernel semi-revision with incision function σ iff it satisfies: inclusion, consistency, core-retainment, pre-expansion and internal exchange.

Proof:

(Construction $\Rightarrow$ Postulates) *Inclusion*, *pre-expansion* and *internal exchange* follow directly from the construction. To prove *core-retainment* assume $\beta \in B \setminus B?\alpha$ then $\beta \in \sigma(B \cup \{\alpha\} \perp\!\!\!\perp \Omega)$, this means that there is $X \in (B \cup \{\alpha\}) \perp\!\!\!\perp \Omega$ such that $\beta \in X$. Consider $B' = X \setminus \{\beta\}$ then $B' \subseteq B$, $Cn(B') \cap \Omega = \emptyset$ and $Cn(B' \cup \{\beta\}) \cap \Omega \neq \emptyset$. To show that *consistency* is satisfied assume by contradiction that it is not. Then $Cn(B?\alpha) \cap \Omega \neq \emptyset$

from compactness it follows that there is $Z \subseteq B?\alpha$ which is finite and such that $Cn(Z) \cap \Omega \neq \emptyset$. We can then infer by monotonicity that there is $Z' \subseteq Z$ such that $Z' \in B \cup \{\alpha\} \perp\!\!\!\perp \Omega$. It follows from the consistency of the logic that $Cn(\emptyset) \cap \Omega = \emptyset$ then we must have that $Z' \neq \emptyset$ and by the construction there must be $\epsilon \in \sigma(B \cup \{\alpha\}) \cap Z'$, but if that is true then $\epsilon \notin B?\alpha$ and $\epsilon \in Z' \subseteq B?\alpha$ which is a contradiction.

(Postulates $\Rightarrow$ Construction) Let ? be an operator satisfying the postulates above and let σ be such that:

$\sigma(B \perp\!\!\!\perp \Omega) = B \setminus \{\beta | \beta \in B?\alpha \text{ for some } \alpha \in B\}$

We have to show (1) that σ is an incision function for the given domain and (2) that $B?\alpha = B?_\sigma\alpha$.

1. First we need to prove that $\sigma(B \perp\!\!\!\perp \Omega) \in \bigcup B \perp\!\!\!\perp \Omega$. Let $\delta \in \sigma(B \perp\!\!\!\perp \Omega)$. It holds that $\delta \notin B?\alpha$ for any $\alpha \in B$. Then it follows from *core-retainment* that there is some $B' \subseteq B$ such that $Cn(B') \cap \Omega = \emptyset$ and $Cn(B' \cup \{\delta\}) \cap \Omega \neq \emptyset$. It follows that there is $B'' \subseteq B'$ such that $B'' \cup \{\delta\} \in B \perp\!\!\!\perp \Omega$.
   Now we have to prove that if $\emptyset \neq X \in B \perp\!\!\!\perp \Omega$ then $X \cap \sigma(B \perp\!\!\!\perp \Omega) \neq \emptyset$. Suppose by contradiction that this is not the case. Then $X \in \{\beta | \beta \in B?\alpha \text{ for some } \alpha \in B\}$. By *internal exchange* we have that $X \subseteq B?\alpha$ for some particular α. Since $Cn(X) \cap \Omega \neq \emptyset$ by monotonicity $Cn(B?\alpha) \cap \Omega \neq \emptyset$ and that contradicts the *consistency*.
2. $\sigma(B \cup \{\alpha\} \perp\!\!\!\perp \Omega) = (B \cup \{\alpha\}) \setminus \{\beta | \beta \in B \cup \{\alpha\}?\epsilon \text{ for some } \alpha \in B\}$ $= (B \cup \{\alpha\}) \setminus (B \cup \{\alpha\})?\alpha$ by *internal exchange* $= (B \cup \{\alpha\}) \setminus B?\alpha$ by *pre-expansion*. Hence, $B?_\sigma\alpha = B \cup \{\alpha\} \setminus \sigma(B \cup \{\alpha\} \perp\!\!\!\perp \Omega) = B \cup \{\alpha\} \setminus ((B \cup \{\alpha\}) \setminus B?\alpha)$ by definition $= B?\alpha$ by *inclusion*
   Hence $B \cup \{\alpha\} \setminus \sigma(B \cup \{\alpha\} \perp\!\!\!\perp \Omega) = B?\alpha$ by *inclusion*.

Representation Theorem 4 [Partial Meet Semi-revision](adapted from [Han97]) The operation ? is a partial meet semi-revision iff it satisfies: inclusion, consistency, relevance, pre-expansion and internal exchange.

Proof:

(Construction $\Rightarrow$ Postulates) *Inclusion*, *pre-expansion* and *internal exchange* follow directly from construction. To prove *relevance* notice that the logic is consistent $Cn(\emptyset) \cap \Omega = \emptyset$ and by lemma 1 $\gamma(B \cup \{\alpha\} \perp \Omega) \neq \emptyset$. By construction we have that if $\beta \in B \setminus B?\alpha$ then there is $B' \in \gamma(B \cup \{\alpha\} \perp \Omega)$ such that $\beta \notin B'$ and by the definition we have that $Cn(B') \cap \Omega = \emptyset$, $\Omega \cap Cn(B' \cup \{\beta\})$ and $\bigcap \gamma(B \cup \{\alpha\} \perp \Omega) \subseteq B' \subseteq B \cup \{\alpha\}$. In order to show that *consistency* is satisfied notice that since the logic is consistent by lemma 1 we have that $B \cup \{\alpha\} \perp \Omega \neq \emptyset$ then by definition $\emptyset \neq \gamma(B \cup \{\alpha\} \perp \Omega) \subseteq B \cup \{\alpha\} \perp \Omega$. Let $X \in B \cup \{\alpha\} \perp \Omega$ then $\Omega \cap Cn(X) = \emptyset$ and $\bigcap \gamma(B \cup \{\alpha\} \perp \Omega) \subseteq X$ and by monotonicity $\Omega \cap Cn(\bigcap \gamma(B \cup \{\alpha\} \perp \Omega) = \emptyset$.

(Postulates $\Rightarrow$ Construction) Let ? satisfies the postulates above and let γ be:

$\gamma = \{X \in B\bot\Omega | B?\alpha \subseteq X \text{ for some } \alpha \in B\}$

We have to show that 1) γ is a selection function and 2) $B?_\gamma\alpha = B?\alpha$

1. We just need to show that $\gamma(B\bot\Omega) \neq \emptyset$. It follows from *consistency* and upper bound property (lemma 2) that there is $X \in B\bot\Omega$ such that $B?\alpha \subseteq X$ and then $X \in \gamma(B\bot\Omega)$
2. $B?_\gamma\alpha = \bigcap\gamma(B \cup \{\alpha\}\bot\Omega) = \bigcap\{X \in B \cup \{\alpha\}\bot\Omega | (B \cup \{\alpha\})?\beta \subseteq X$ for some $\beta \in B\}$ = by *internal exchange* $\bigcap\{X \in B \cup \{\alpha\}\bot\Omega | (B \cup \{\alpha\})?\alpha \subseteq X\}$ = by *pre-expansion* $\bigcap\{X \in B \cup \{\alpha\}\bot\Omega | B?\alpha \subseteq X\}$. Hence $B?\alpha \subseteq B?_\sigma\alpha$.

   Now let $\beta \notin B?\alpha$. If $\beta \notin B \cup \{\alpha\}$ then $\beta \notin B?_\sigma\alpha$. Suppose that $\beta \in B\cup\{\alpha\}$. By *pre-expansion* $\beta \notin (B+\alpha)?\alpha$. Since $\beta \in B+\alpha \backslash (B+\alpha)?\alpha$ by *relevance* and *pre-expansion* we have that there is B' such that $B?\alpha \subseteq B' \subseteq B\cup\{\alpha\}$, $\Omega\cap Cn(B') = \emptyset$ and $\Omega \cap Cn(B'\cup\{\beta\} \neq \emptyset$ then by *upper bound* property (lema 2) there is X such that $B' \subseteq X \in (B+\alpha)\bot\Omega$. Since $B?\alpha \subseteq X$ we have that $X \in \gamma((B+\alpha)\bot\Omega)$ and since $\beta \notin X$, $\beta \notin (B+\alpha)?_\sigma\alpha$ by *pre-expansion* we have $B?_\sigma\alpha \subseteq B?\alpha$

Representation Theorem 5 [Partial meet revision with weak-success] The operator $*$ is a partial meet revision without negation with weak success if and only if it satisfies the following postulates for revision: weak-success, consistency, inclusion, relevance and pre-expansion.

Proof:

(Construction $\Rightarrow$ Postulates) *Inclusion*, *pre-expansion* and *consistency* follow from the construction. To prove *success* assume that $Cn(\{\alpha\})\cap\Omega = \emptyset$ then $K * \alpha = \bigcap\gamma(K \cup \{\alpha\}\bot\Omega, \alpha)$. We need to show that there is $X \in K \cup \{\alpha\}\bot\Omega$ such that $\alpha \in X$. The existence of such X follows from $Cn(\{\alpha\})$ and the lemma 2. In order to show that *relevance* is satisfied, assume $\beta \notin \bigcap\gamma(K \cup \{\alpha\}\bot\Omega, \alpha)$ then there is $K' \in \gamma(K \cup \{\alpha\}\bot\Omega, \alpha$ with $\beta \notin K'$. By definition $\Omega \cap Cn(K') = \emptyset$, $\Omega \cap Cn(K' \cup \{\beta\}) \neq \emptyset$ and $\bigcap\gamma(K \cup \{\alpha\}\bot\Omega) \subseteq K' \subseteq K + \alpha$.

(Postulates $\Rightarrow$ Construction) Let $*$ be an operator satisfying the postulates above and let:

$\gamma(K \cup \{\alpha\}\bot\Omega, \alpha) = \{X \in K \cup \{\alpha\}\bot\Omega : K * \alpha \subseteq X\}$

We need to prove that 1) γ is a selection function and 2) $K *_\gamma \alpha = K * \alpha$

1. To prove that γ is a selection function we need to show that $\gamma(K \cup \{\alpha\}\bot\Omega, \alpha) \neq \emptyset$. We have that $\Omega\cap Cn(K*\alpha) = \emptyset$ by *consistency* and $K*\alpha \subseteq K+\alpha$ by *inclusion*, then by lemma 2 there is $K*\alpha \subseteq K' \in (K\cup\{\alpha\})\bot\Omega$

2. To prove that $K * \alpha = \bigcap \gamma(K \cup \{\alpha\} \bot \Omega, \alpha)$ notice that $\emptyset \neq K * \alpha \subseteq X$ for all $X \in K \cup \{\alpha\} \bot \Omega$ and then $K * a \subseteq \bigcap \gamma(K \cup \{\alpha\} \bot \Omega, \alpha)$ $\bigcap \gamma(K \cup \{\alpha\} \bot \Omega, \alpha) \subseteq K * \alpha$ follows from *relevance* and $\alpha \in \bigcap \gamma(K \cup \{\alpha\} \bot \Omega, \alpha)$ if $\Omega \cap Cn(\{\alpha\}) = emptyset$ follows from *success*

Representation Theorem 6 [Kernel revision with weak-success]
The operator ? is a kernel revision without negation with weak success if and only if it satisfies the following postulates for revision: weak-success, consistency, inclusion, core-retainment and pre-expansion.

Proof:

(Construction $\Rightarrow$ Postulates) Let $?_\sigma$ be an operator of kernel revision without negation with weak success based on an incision function that almost protects the input, σ. It follows directly from the construction that *inclusion* and *pre-expansion* are satisfied. From the definition of an incision function that protects the consistent inputs, it follows that $?_\sigma$ satisfies *weak success* and *consistency*. Finally, for *core-retainment*, let $\beta \in B \setminus B?_\sigma\alpha$. Then by construction $\beta \in \sigma(\alpha, (B \cup \{\alpha\}) \perp\!\!\!\perp \Omega)$. This means that for some set $X \in (B \cup \{\alpha\}) \perp\!\!\!\perp \Omega$, $\beta \in X$. Let $B' = X \setminus \{\beta\}$. We have $B' \subseteq B \cup \{\alpha\}$ since X is minimal then $Cn(B') \cap \Omega = \emptyset$, but $Cn(B' \cup \{\beta\}) \cap \Omega \neq \emptyset$.

(Postulates $\Rightarrow$ Construction) Let ? be an operator satisfying the postulates above and let σ be such that for every formula α:

$\sigma(\alpha, B \perp\!\!\!\perp \Omega) = B \setminus (B?\alpha)$

We have to show (1) that σ is an incision function that almost protects the input for the given domain and (2) that $B?\alpha = B?_\sigma\alpha$.

1. We have to show that the three conditions of Definition 11 are satisfied. For the first condition, let $\beta \in \sigma(\alpha, B \perp\!\!\!\perp \Omega)$. Then it holds that $\beta \in B \setminus (B?\alpha)$ and it follows from *core-retainment* that there is some $B' \subseteq B+\alpha$ such that $\Omega \cap Cn(B') = \emptyset$ and $\Omega \cap Cn(B' \cup \{\beta\}) \neq \emptyset$. It follows that there is some subset B'' of B' such that $B'' \cup \{\beta\} \in B \perp\!\!\!\perp \Omega$ and hence, $\beta \in \bigcup(B \perp\!\!\!\perp \Omega)$.
   For the second condition, let $\emptyset \neq X \in B \perp\!\!\!\perp \Omega$. Suppose that $X \cap \sigma(\alpha, B \perp\!\!\!\perp \Omega) = \emptyset$. Then $X \subseteq B?\alpha$. Since $\Omega \cap Cn(X) \neq \emptyset$, it follows from monotony that $\Omega \cap Cn(B?\alpha) \neq \emptyset$, contrary to *consistency*. This contradiction is sufficient to prove that $X \cap \sigma(\alpha, B \perp\!\!\!\perp \Omega) \neq \emptyset$.
   For the third condition, suppose $\Omega \cap Cn(\{\alpha\}) = emptyset$. By *weak success*, $\alpha \in B?\alpha$, and hence, $\alpha \notin \sigma(\alpha, B \perp\!\!\!\perp \Omega)$.
2. By definition, $\sigma(\alpha, (B \cup \{\alpha\}) \perp\!\!\!\perp \Omega) = (B \cup \{\alpha\}) \setminus ((B \cup \{\alpha\})?\alpha) = (B \cup \{\alpha\}) \setminus B?\alpha$ (*pre-expansion*). Hence, $B?_\sigma\alpha = (B \cup \{\alpha\}) \setminus \sigma((B \cup \{\alpha\}) \perp\!\!\!\perp \Omega) = (B \cup \{\alpha\}) \setminus ((B \cup \{\alpha\}) \setminus B?\alpha) = B?\alpha$ (*inclusion*). $\square$

Representation Theorem 7 [Partial meet revision with success]
The operator $*$ is a partial meet revision with weak success if and only if it satisfies the following postulates for revision: success, weak-consistency, inclusion, relevance and pre-expansion.

Proof:

(Construction $\Rightarrow$ Postulates) *Inclusion*, *pre-expansion* and *weak-consistency* follows directly from the construction. To prove *success* and *relevance* we analyze two cases: If $Cn(\{\alpha\}) \cap \Omega = \emptyset$ then success and relevance is satisfied by the same reason presented in the demonstration of representation theorem 5. If $Cn(\{\alpha\} \cap \Omega \neq \emptyset$ then success is satisfied because $\alpha \in B * \alpha = B + \alpha$, relevance is satisfied because there is no $\beta \in B \setminus B + \alpha$.

(Postulates $\Rightarrow$ Construction) Let $*$ be an operator satisfying the postulates above and let:

$\gamma(K \cup \{\alpha\} \perp \Omega, \alpha) = \{X \in K \cup \{\alpha\} \perp \Omega : K * \alpha \subseteq X\}$ if $\Omega \cap Cn(\{\alpha\}) = \emptyset$

$\gamma(K \cup \{\alpha\} \perp \Omega, \alpha) = \{K \cup \{\alpha\}\}$ if $\Omega \cap Cn(\{\alpha\}) \neq \emptyset$

We need to prove that 1) γ is a selection function and 2) $K *_\gamma \alpha = K * \alpha$

1. If $\Omega \cap Cn(\{\alpha\}) = \emptyset$ we have $\gamma(K + \cup\{\alpha\} \perp \Omega, \alpha) \neq \emptyset$ notice that $\Omega \cap Cn(K * \alpha) = \emptyset$ by *consistency* and $K * \alpha \subseteq K + \alpha$ by *inclusion*, then by *upper bound property* there is $K * \alpha \subseteq K' \in (K \cup \{\alpha\}) \perp \Omega$.
2. To prove that $K * \alpha = \bigcap \gamma(K \cup \{\alpha\} \perp \Omega, \alpha)$ notice that $\emptyset \neq K * \alpha \subseteq X$ for all $X \in (K \cup \{\alpha\}) \perp \Omega$ and then $K * \alpha \subseteq \bigcap \gamma(K \cup \{\alpha\} \perp \Omega, \alpha)$
   If α is consistent then $\bigcap \gamma(K \cup \{\alpha\} \perp \Omega, \alpha) \subseteq K * \alpha$ follows from *relevance* and $\alpha \in \bigcap \gamma(K \cup \{\alpha\} \perp \Omega, \alpha)$ follows from *success*.
   If $\Omega \cap Cn(\{\alpha\}) \neq \emptyset$ then since $K * a$ satisfies *success*, *inclusion* and *relevance* it satisfies *inconsistent expansion* and then $K * a = K + a$.

Representation Theorem 8 [Kernel Revision with Success] The operator $*$ is a kernel revision without negation if and only if it satisfies the following postulates for revision: success, weak-consistency, inclusion, core-retainment, pre-expansion.

Proof:

(Construction $\Rightarrow$ Postulates) Let $*_\sigma$ be an operator of kernel revision without negation based on an incision function σ that protects the input. It follows directly from the construction that *inclusion* and *pre-expansion* are satisfied. From the definition of an incision function that protects the input it follows that $*_\sigma$ satisfies *success* and *weak consistency*. Finally, for *core-retainment*, let $\beta \in B \setminus B *_\sigma \alpha$. Then by construction $\beta \in \sigma(\alpha, (B \cup \{\alpha\}) \perp\!\!\!\perp \Omega)$. This means that for some set $X \in (B \cup \{\alpha\}) \perp\!\!\!\perp \Omega$, $\beta \in X$. Let $B' = X \setminus \{\beta\}$. We have $B' \subseteq B \cup \{\alpha\}$, $\Omega \cap Cn(B') = \emptyset$ and $\Omega \cap Cn(B' \cup \{\beta\}) \neq \emptyset$.

(Postulates ⇒ Construction) Let $*$ be an operator satisfying the postulates above and let σ be such that for every formula α:

$\sigma(\alpha, B \perp\!\!\!\perp \Omega) = B \setminus (B * \alpha)$

We have to show (1) that σ is an incision function that protects the input for the given domain and (2) that $B * \alpha = B *_\sigma \alpha$.

1. We have to show that the three conditions of Definition 13 are satisfied. For the first condition, let $\beta \in \sigma(\alpha, B \perp\!\!\!\perp \Omega)$. Then it holds that $\beta \in B \setminus (B * \alpha)$ and it follows from *core-retainment* that there is some $B' \subseteq B + \alpha$ such that $\Omega \cap Cn(B') = \emptyset$ and $\Omega \cap Cn(B' \cup \{\beta\}) \neq \emptyset$. It follows that there is some subset B'' of B' such that $B'' \cup \{\beta\} \in B \perp\!\!\!\perp \Omega$ and hence, $\beta \in \bigcup(B \perp\!\!\!\perp \Omega)$.
   For the second condition, let $\Omega \cap Cn(\{\alpha\}) = \emptyset$ and $\emptyset \neq X \in B \perp\!\!\!\perp \Omega$. Suppose that $X \cap \sigma(\alpha, B \perp\!\!\!\perp \Omega) = \emptyset$. Then $X \subseteq B * \alpha$. Since $\Omega \cap Cn(X) \neq \emptyset$, it follows from monotony that $\Omega \cap Cn(B * \alpha) \neq \emptyset$, contrary to *weak consistency*. This contradiction is sufficient to prove that $X \cap \sigma(\alpha, B \perp\!\!\!\perp \Omega) \neq \emptyset$.
   For the third condition, it suffices to note that by *success*, $\alpha \in B * \alpha$, and hence, $\alpha \notin \sigma(\alpha, B \perp\!\!\!\perp \Omega)$.
2. By definition, $\sigma(\alpha, (B \cup \{\alpha\}) \perp\!\!\!\perp \Omega) = (B \cup \{\alpha\}) \setminus ((B \cup \{\alpha\}) * \alpha) = (B \cup \{\alpha\}) \setminus B * \alpha$ (*pre-expansion*). Hence, $B *_\sigma \alpha = (B \cup \{\alpha\}) \setminus \sigma((B \cup \{\alpha\}) \perp\!\!\!\perp \Omega) = (B \cup \{\alpha\}) \setminus ((B \cup \{\alpha\}) \setminus B * \alpha) = B * \alpha$ (*inclusion*). □

B Other Theorems

In section 5 we have mentioned that if we get the minimal kernels of B w.r.t. each element of A then we have $B \perp\!\!\!\perp A$. We are going to prove that. First we have to define an operator that given a finite set of sets returns just the minimal ones:

Definition 14 *Let A be a finite set of sets. $min(A) = \{x \in A | \forall z \in A(z \subseteq x \Rightarrow z = x)\}$*

Theorem 1 $B \perp\!\!\!\perp A = min(\bigcup\{B \perp\!\!\!\perp \alpha | \alpha \in A\})$

Proof: If $X \in B \perp\!\!\!\perp A$ then $Cn(X) \cap A \neq \emptyset$. Let $\alpha \in Cn(X) \cap A$ then for all $Y \subset X$ we have $\alpha \notin Cn(Y)$, thus, $X \in B \perp\!\!\!\perp \alpha$. Since $X \in B \perp\!\!\!\perp A$, $\nexists Y \subset X$ such that $Y \in B \perp\!\!\!\perp A$, thus, $X \in min(\bigcup\{B \perp\!\!\!\perp \alpha | \alpha \in A\}$. Hence, $B \perp\!\!\!\perp A \subseteq min(\bigcup\{B \perp\!\!\!\perp \alpha | \alpha \in A\}$

If $X \in min(\bigcup\{B \perp\!\!\!\perp \alpha | \alpha \in A\}$ then $X \in B \perp\!\!\!\perp \alpha$ for some $\alpha \in A$. Hence, $Cn(X) \cap A \neq \emptyset$. To prove that for all $Y \subset X$ we have $Cn(Y) \cap A = \emptyset$, assume by contradiction that $\beta \in Cn(Y) \cap A$. Then we have that there is $Y' \subseteq Y$ such that $\beta \in Y'$, and for all $Y'' \subset Y'$ $\beta \notin Cn(Y')$. However, that means that $Y' \in \bigcup\{B \perp\!\!\!\perp \alpha | \alpha \in A\}$. However, $Y' \subseteq Y \subset X$ contradicts $X \notin min(\bigcup\{B \perp\!\!\!\perp \alpha | \alpha \in A\}$. Hence, $min(\bigcup\{B \perp\!\!\!\perp \alpha | \alpha \in A\} \subseteq B \perp\!\!\!\perp A$

References

[AGM85] Carlos Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change. *Journal of Symbolic Logic*, 50(2):510–530, 1985.

[AM81] Carlos Alchourrón and David Makinson. Hierarchies of regulation and their logic. In Hilpinen, editor, *New Studies in Deontic Logic*, pages 125–148. D. Reidel Publishing Company, 1981.

[AM82] Carlos Alchourrón and David Makinson. On the logic of theory change: contraction functions and their associated revision functions. *Theoria*, 48:14–37, 1982.

[AM85] Carlos Alchourrón and David Makinson. On the logic of theory change: Safe contraction. *Studia Logica*, 44:405–422, 1985.

[BCM+03] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, 2003.

[BLHL01] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*, May, 2001.

[BP07] Franz Baader and Rafael Penaloza. Axiom pinpointing in general tableaux. In *Proceedings of the 16th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods TABLEAUX 2007*, LNAI, Aix-en-Provence, France, 2007. Springer.

[DSB+04] Mike Dean, Gus Schreiber, Sean Bechhofer, Frank van Harmelein, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. OWL web onotology language reference. W3C Recomendation, 2004.

[FHP+06] Giorgos Flouris, Zhisheng Huang, Jeff Z. Pan, Dimitris Plexousakis, and Holger Wache. Inconsistencies, negations and changes in ontologies. In *AAAI*, 2006.

[Flo06] Giorgos Flouris. *On Belief Change and Ontology Evolution*. PhD thesis, University of Crete, Department of Computer Science, 2006.

[FPA04] Giorgos Flouris, Dimitris Plexousakis, and Grigoris Antoniou. Generalizing the AGM postulates: preliminary results and applications. In *In Proc. of the 10th International Workshop on Non-Monotonic Reasoning 2004 (NMR-04)*, pages 171–179, Westin Whistler Resort and Spa, Whistler BC, Canada, June 6-8 2004.

[FPA05a] Giorgos Flouris, Dimitris Plexousakis, and Grigoris Antoniou. On applying the AGM theory to DLs and OWL. In *Proceedings of the International Semantic Web Conference*, pages 216–231, 2005.

[FPA05b] Giorgos Flouris, Dimitris Plexousakis, and Grigoris Antoniou. Updating description logics using the AGM theory. In *In Proceedings of the 7th International Symposium on Logical Formalizations of Commonsense Reasoning*, Corfu, Greece, May 2005.

[FPA06] Giorgos Flouris, Dimitris Plexousakis, and Grigoris Antoniou. On generalizing the agm postulates. In *Proceedings of the 3rd European Starting AI Researcher Symposium (STAIRS-06)*, 2006.

[Fuh91] A. Fuhrmann. Theory contraction through base contraction. *Journal of Philosophical Logic*, 20:175–203, 1991.

[Gal92] Julia Rose Galliers. Autonomous belief revision and communication. In Peter Gärdenfors, editor, *Belief Revision*, volume 29 of *Cambridge Tracts in Theoretical Computer Science*, pages 220–246. Cambridge University Press, 1992.

[Gär88] Peter Gärdenfors. *Knowledge in Flux - Modeling the Dynamics of Epistemic States*. MIT Press, 1988.

[GR95] Peter Gärdenfors and Hans Rott. Belief revision. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume IV, chapter 4.2. Oxford University Press, 1995.

[Han91] Sven Ove Hansson. *Belief Base Dynamics*. PhD thesis, Uppsala University, 1991.

[Han92a] Sven Ove Hansson. A dyadic representation of belief. In Peter Gärdenfors, editor, *Belief Revision*, volume 29 of *Cambridge Tracts in Theoretical Computer Science*, pages 89–121. Cambridge University Press, 1992.

[Han92b] Sven Ove Hansson. Reversing the Levi identity. *Journal of Philosophical Logic*, 22:637–639, 1992.

[Han94] Sven Ove Hansson. Kernel contraction. *Journal of Symbolic Logic*, 59:845–859, 1994.

[Han97] Sven Ove Hansson. Semi-revision. *Journal of Applied Non-Classical Logic*, 7(1-2):151–175, 1997.

[Han99a] Sven Ove Hansson. A survey of non-prioritized belief revision. *Erkenntnis*, 50(2/3):413–427, 1999.

[Han99b] Sven Ove Hansson. *A Textbook of Belief Dynamics*. Kluwer Academic Press, 1999.

[HPS03] I. Horrocks and P. Patel-Schneider. Reducing OWL entailment to description logic satisfiability. In *Proc. of the 2nd International Semantic Web Conference (ISWC)*, 2003.

[HPS04] Ian Horrocks and Peter Patel-Schneider. Reducing OWL entailment to description logic satisfiability. *Journal of Web Semantics*, 1(4):345–357, 2004.

[HS04] Peter Haase and York Sure. State-of-the-art on ontology evolution. SEKT informal deliverable 3.1.1.b, Institute AIFB, University of Karlsruhe, 2004.

[HST99] I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for expressive description logics. In H. Ganzinger, D. McAllester, and A. Voronkov, editors, *Proc. of the 6th Int. Conf. on Logic for Programming and Automated Reasoning (LPAR'99)*, number 1705 in Lecture Notes in Artificial Intelligence, pages 161–180. Springer, 1999.

[HW02] Sven Ove Hansson and Renata Wassermann. Local change. *Studia Logica*, 70(1):49–76, 2002.

[HWKP06] Christian Halaschek-Wiener, Yarden Katz, and Bijan Parsia. Belief base revision for expressive description logics. In *OWL: Experiences and Directions 2006*, 2006.

[Kal06] Aditya Kalyanpur. *Debugging and repair of owl ontologies*. PhD thesis, Faculty of the Graduate School of the University of Maryland, 2006.

[KL07] Seung Hwan Kang and Sim Kim Lau. Ontology revision on the semantic web: Integration of belief revision theory. In *HICSS '07: Proceedings of the 40th Annual Hawaii International Conference on System Sciences*. IEEE Computer Society, 2007.

[KPS05] Aditya Kalyanpur, Bijan Parsia, and Evren Sirin. Black box techniques for debugging unsatisfiable concepts. In Ian Horrocks, Ulrike Sattler, and Frank Wolter, editors, *Description Logics*, volume 147 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2005.

[KPSH05] Aditya Kalyanpur, Bijan Parsia, Evren Sirin, and James A. Hendler. Debugging unsatisfiable classes in owl ontologies. *J. Web Sem.*, 3(4):268–293, 2005.

[Mak97] David Makinson. Screened revision. *Theoria*, 63:14–23, 1997.

[MLBP06] Thomas Meyer, Kevin Lee, Richard Booth, and Jeff Z. Pan. Finding maximally satisfiable terminologies for the description logic ALC. In *AAAI*. AAAI Press, 2006.

[MvH04] Deborah L. McGuinness and Frank van Harmelen. OWL Web Ontology Language overview. W3c recommendation, World Wide Web Consortium, February 2004.

[Neb90] Bernhard Nebel. *Reasoning and Revision in Hybrid Representation Systems*, volume 422 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 1990.

[New82] Allen Newell. The knowledge level. *Artificial Intelligence*, 18(1):87–127, 1982.

[NM01] Natalya F. Noy and Deborah L. McGuinness. Ontology development 101: A guide to creating your first ontology. Technical Report KSL-01-05, Knowledge Systems Laboratory, Stanford University, Stanford, CA, 94305, USA, MAR 2001.

[Rei87] Raymond Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–95, 1987.

[RW06] Márcio Moretto Ribeiro and Renata Wassermann. First steps towards revising ontologies. In *Proceedings of the Second Workshop on Ontologies and their Applications (WONTO 2006)*, 2006.

[SC03] Stefan Schlobach and Ronald Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In Georg Gottlob and Toby Walsh, editors, *IJCAI*, pages 355–362. Morgan Kaufmann, 2003.

[Was00] Renata Wassermann. An algorithm for belief revision. In *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR2000)*. Morgan Kaufmann, 2000.