

Pseudo-contraction Operations for Description Logics

Vinícius Bitencourt Matos

MASTER'S THESIS SUBMITTED TO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
OF
UNIVERSIDADE DE SÃO PAULO

Master's Programme: Computer Science
Supervisor: Prof. Renata Wassermann, PhD

The author has received funding from CNPq (grant 131803/2018-2).

São Paulo, July 2021

Pseudo-contraction Operations for Description Logics

This version of the thesis incorporates the corrections and changes which have been suggested by the examination committee during the presentation of the original version, which took place on 13/07/2021. A copy of the original version is available at IME-USP.

Examination Committee:

- Prof. Renata Wassermann, PhD (supervisor) — IME-USP
- Prof. Frederico Luiz Gonçalves de Freitas, PhD — CIn-UFPE
- Prof. Eduardo Leopoldo Fermé, PhD — Universidade da Madeira (Portugal)

Pseudo-contraction Operations for Description Logics

Abstract

MATOS, V. B. **Pseudo-contraction Operations for Description Logics**. 2021. 62 pp. Master's Thesis – Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2021.

Knowledge representation in ontologies is based on Description Logics, which are decidable fragments of first-order logic. Since knowledge is not static, it is necessary to deal with the acquisition of new information, which may contradict the existing knowledge. Belief Revision aims to solve this problem, but the classical AGM framework assumes an ideal agent that is able to deal with logically closed sets of sentences, and some of its generalisations for belief bases (such as ontologies represented in Description Logics) may lead to loss of information due to the fact that no sentence can be added when a contraction operation is performed. In this work, we analyse kernel constructions for pseudo-contraction operations and their formal properties. Also, we show the close relationship between concepts and definitions of Belief Revision and Ontology Repair (such as pseudo-contractions and gentle repairs, respectively), and we propose a unified notation for their operations.

Keywords: Belief Revision, Pseudo-contraction, Ontology Repair, Description Logics.

Resumo

MATOS, V. B. **Operações de Pseudocontração para Lógicas de Descrição**. 2021. 62 pp. Dissertação de Mestrado — Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2021.

A representação de conhecimento em ontologias é baseada em Lógicas de Descrição, que são fragmentos decidíveis da lógica de primeira ordem. Uma vez que o conhecimento não é estático, é necessário lidar com a aquisição de novas informações, que podem contradizer o conhecimento existente. Revisão de Crenças busca resolver este problema, mas o arcabouço clássico AGM supõe um agente ideal capaz de lidar com conjuntos de sentenças fechados logicamente, e algumas de suas generalizações para bases de crenças (como ontologias representadas com Lógicas de Descrição) podem levar a uma perda de informação, visto que nenhuma sentença pode ser adicionada ao realizar uma operação de contração. Neste trabalho, analisamos construções *kernel* para operações de pseudocontração e suas propriedades formais. Ademais, mostramos a relação próxima que existe entre os conceitos e definições das áreas de Revisão de Crenças e Reparo de Ontologias (tais como pseudocontrações e reparos brandos, respectivamente), e apresentamos uma notação unificada para suas operações.

Palavras-chave: Revisão de Crenças, Pseudocontração, Reparo de Ontologias, Lógicas de Descrição.

Contents

List of Figures	xi
List of Tables	xiii
1 Introduction	1
1.1 Objectives	2
1.2 Organisation of this text	2
2 Background	3
2.1 Description Logics	3
2.1.1 Syntax and semantics	4
2.1.2 Some usual Description Logics	6
2.1.3 OWL	8
2.2 Belief Revision	8
2.2.1 AGM Framework	9
2.2.2 Generalisation for belief bases	11
3 Pseudo-contraction Operations	15
3.1 Cn^* partial meet pseudo-contraction	16
3.2 Cn^* kernel pseudo-contraction	20
3.3 Pseudo-contraction plug-in for Protégé: a prototype	24
3.4 Other operations	28
3.4.1 Two-place pseudo-contractions	28
3.4.2 Pseudo-revisions and adapted identities	30
4 Pseudo-contractions and gentle repairs	33
4.1 Ontology Repair	33
4.2 Correspondence between Belief Revision and Repairs in Description Logics	36
5 Conclusion and future work	43
Bibliography	45

List of Figures

3.1	Screenshot of Protégé showing some types of consequences	16
3.2	Screenshot of Protégé showing the menu with the option to insert the new tab	25
3.3	Screenshot of Protégé with the pseudo-contraction tab	25
3.4	Screenshot of Protégé showing the kernel set of an example	26
3.5	Screenshot of Protégé showing the remainder set of an example	26
3.6	Screenshot of Protégé showing the remainder set of an example that uses Cn^*	27
3.7	Screenshot of Protégé showing the kernel set of an example that uses Cn^*	27
3.8	‘Subtype’ relations among Belief Revision (pseudo-)contraction operations	29
4.1	Relationship between concepts in Belief Change and Ontology Repair	39

List of Tables

2.1	ABox axiom types.	4
2.2	TBox axiom types.	4
2.3	RBox axiom types.	5
2.4	Concept constructors in \mathcal{AL}	6
2.5	Some role constructors.	6
2.6	Some concept constructors.	7
2.7	OWL and OWL 2 profiles.	8

Chapter 1

Introduction

In computer science, *ontologies* are shareable representations of a domain’s knowledge. An ontology represents concepts of a domain, its individuals and binary relations between them. Ontology development and maintenance tasks involve ontology engineers, domain specialists and other professionals. In this collaborative process, one complicating aspect is that even a small modification may impact considerably on what the ontology entails. To facilitate the execution of those tasks, techniques to aid in repairing and evolving ontologies (and represented knowledge in general) have been created. *Ontology change* is the generic problem of making changes to an ontology in response to a certain requirement, and covers several specific problems such as *ontology debugging*, *ontology merging* and *ontology evolution* [FMK⁺08].

The area of *Ontology Repair* groups together a set of formal definitions and tools devised to help ontology maintainers in the task of debugging and getting rid of unwanted inferences. Different approaches have been proposed, depending on whether one is interested in repairing only the ABox [LLR⁺10, DQ15], i.e., the part of the ontology dealing with instances, while leaving the terminological part (TBox) fixed, or considering the ontology as a whole [Kal06, JHQ⁺09, Hor11].

Belief Revision is a research area that aims at solving problems related to changing knowledge bases or logical theories, especially in the face of new, possibly conflicting, information. The work of Alchourrón, Gärdenfors and Makinson [AGM85] is widely recognised as the initial hallmark of this area of research, and gave rise to what is known as the *AGM paradigm*. Initially developed having propositional logic in mind, in the last decades the AGM theory has been adapted to several other formalisms, including Description Logics [FPA05, RW09, Was11]. Therefore, we can model problems in ontology maintenance using this framework (with a few modifications).

Both in Belief Revision and in Ontology Repair, classical approaches assume that nothing should be added to a knowledge base when we perform the task of removing some unwanted information. Whilst this may be a reasonable assumption, it is usually formalised in a way that forces us to discard too much information: the sentence ‘*s* is a European white swan’, for example, must be completely removed if we no longer want the information that ‘*s* is white’, and thus we lose all the knowledge it represented. In classical Belief Revision, an agent’s knowledge is an infinite set of logical sentences that is logically closed, which means that every consequence of the set is also in the set; thus, in our example, the agent would also have the sentences ‘*s* is European’ and ‘*s* is a swan’, which could be preserved after the removal of ‘*s* is a European white swan’. Therefore, the aforementioned assumption is reasonable for closed sets, but it can be too restrictive for belief bases (which do not require logical closure and may be finite, such as ontologies). Nonetheless, it should not be disregarded altogether, as it prevents the addition of unrelated knowledge (e.g. if we want to remove ‘*s* is white’, this does not mean that we should allow the addition of ‘*s* is yellow’, ‘*s* is Brazilian’ or ‘*s* is an aircraft’). That assumption can be replaced with a less restrictive version which only states that we cannot add new *consequences* to the knowledge base, thus allowing to add sentences that were logically entailed by the original set. In the last decades, this idea has been proposed and developed both in Belief Revision and in Ontology Repair [Han93a, Mey01, RW08b, LSPV08, San16, BKNP18, TCG⁺18], with different terminologies and notations. In particular, pseudo-contractions are generalisations of

contractions in Belief Revision which allow the addition of sentences as long as they were already entailed by the initial set, and they can be obtained from adaptations of partial meet and kernel constructions for contraction. In a partial meet construction, the contraction result consists of sentences that are in maximal ‘good’ fragments of the original set (by ‘good’ we mean ‘not implying the sentence to be removed’). A kernel contraction is obtained by removing from the original set some sentences that contribute to the entailment of the unwanted sentence so as to break minimal ‘bad’ fragments of the set (where ‘bad’ means ‘implying the sentence that should be removed’). In a partial meet or kernel pseudo-contraction, we allow the original set to ‘grow’ with the addition of some of its logical consequences before contracting. Similarly, a gentle repair of an ontology is built by removing sentences or replacing them with weaker versions so that the resulting set does not imply the unwanted sentence, and no new consequence is allowed.

1.1 Objectives

The objectives of this research are the following:

- Prove a characterisation and derive formal properties of a new pseudo-contraction operation that is based on a kernel construction.
- Contribute towards the integration of Ontology Repair (a specific type of ontology change) and Belief Revision by showing the relationship between their concepts and operations, such as gentle repairs and pseudo-contractions.

1.2 Organisation of this text

The organisation of this text is as follows. [Chapter 2](#) introduces the research areas and their concepts: in [Section 2.1](#), we present the definition of logic, consequence operator and Description Logics, showing the syntax and semantics of Description Logic concepts, roles and sentences, some usual Description Logics and the family of OWL languages; [Section 2.2](#) contains an introduction to the area of Belief Revision, its operations, postulates and constructions, primarily focusing on contraction operations. [Chapter 3](#)¹ covers pseudo-contraction operations and some of its constructions — namely, Cn^* partial meet pseudo-contraction and Cn^* kernel pseudo-contraction — and a prototype of a plug-in for an ontology editor that computes the pseudo-contractions. In [Chapter 4](#)², we introduce the general problem of ontology change and show that pseudo-contraction operations (from Belief Revision) and gentle repairs (from Ontology Repair) are closely related, and many of their ideas and definitions are equivalent or remarkably similar. [Chapter 5](#) finishes the text with the conclusion and a few ideas for future work.

¹Parts of [Chapter 3](#) appeared in [SMRW18].

²Parts of [Chapter 4](#) were published in [MGSW19].

Chapter 2

Background

We will denote a language by \mathcal{L} , and we will use Cn to refer explicitly to a consequence operator. In this way, $\text{Cn}(X)$ denotes the logical consequences of Cn over X , where X is a set of formulae in \mathcal{L} . As we will be dealing with a family of logics (Description Logics), we will assume that in each case Cn is associated to the smallest Description Logic that satisfies some required properties and is expressive enough to be able to represent the set of formulae given as argument. Below we list some properties a consequence relation may have.

- **monotonicity:** if $X \subseteq Y$, then $\text{Cn}(X) \subseteq \text{Cn}(Y)$;
- **inclusion:** $X \subseteq \text{Cn}(X)$;
- **idempotence:** $\text{Cn}(X) = \text{Cn}(\text{Cn}(X))$;
- **supraclassicality:** $\text{Cn}_{\text{PL}}(X) \subseteq \text{Cn}(X)$, where Cn_{PL} is the consequence relation of propositional logic;
- **compactness:** for any X , if $\alpha \in \text{Cn}(X)$, there is a finite $X' \subseteq X$ such that $\alpha \in \text{Cn}(X')$;
- **subclassicality** (w.r.t. $\text{Cn}_{\text{CLASSICAL}}$): $\text{Cn}(X) \subseteq \text{Cn}_{\text{CLASSICAL}}(X)$, where $\text{Cn}_{\text{CLASSICAL}}$ is a classical consequence operator (which is usually Tarskian — such as Cn_{FOL} , the consequence relation of first-order logic).

If a consequence relation satisfies monotonicity, inclusion and idempotence, we say that it is Tarskian.

2.1 Description Logics

Description Logics (henceforth, DLs) are a family of decidable fragments of first-order logic that can be used to represent structured knowledge about a domain [BHL17]. While first-order logic could be directly used to represent such knowledge, the fact that this logic is undecidable implies that it is impossible to write an algorithm that always decides (in a finite number of steps) whether a given sentence is entailed by a set of sentences. The use of decidable, less expressive logics makes it possible to create automatic reasoners that will always stop and produce the correct answer.

In a Description Logic, *individuals* represent elements of a given domain, *concepts* (or *classes*) represent sets of individuals and *roles* (or *properties*) represent sets of pairs of individuals. Individuals, concepts and roles correspond to constants, unary predicates and binary predicates in first-order logic, respectively. The signature of a Description Logic is a triple $\langle N_C, N_R, N_I \rangle$ consisting of mutually disjoint sets of concept names (N_C), role names (N_R) and individual names (N_I). Concept names and role names are also called atomic concepts and atomic roles, respectively. Complex concepts and roles can be obtained from atomic concepts and roles by constructors, and each Description Logic has its set of allowed constructors.

A knowledge base in a Description Logic consists of terminological, assertional and relational knowledge:

- the *TBox*, which contains the terminological knowledge about the concepts, their definitions and hierarchy;
- the *ABox*, which has assertions about individuals;
- the *RBox*, which contains definitions and hierarchy of roles. This is only allowed by some expressive Description Logics.

2.1.1 Syntax and semantics

To define the semantics of Description Logics, we will consider *interpretations* $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, which consist of a non-empty set $\Delta^{\mathcal{I}}$ (the domain) and an interpretation function $\cdot^{\mathcal{I}}$, which assigns, for every $A \in N_C$, $r \in N_R$ and $a \in N_I$, respectively, a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ of individuals, a subset $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ of pairs of individuals, and an individual $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ [BN03].

Tables 2.2, 2.1 and 2.3 show the types of axioms and their corresponding semantics. In the following, A represents atomic concepts, $r_{(i)}$ and s represent atomic roles, t represents (atomic or complex) roles, $a_{(i)}$ and b represent individuals, and both C and D represent (atomic or complex) concepts.

Table 2.1: *ABox axiom types.*

Name	Notation	Semantics
concept assertion	$C(a)$ or $a : C$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
role assertion	$t(a, b)$ or $a t b$	$\langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in t^{\mathcal{I}}$

Table 2.2: *TBox axiom types.*

Name	Notation	Semantics
general concept inclusion	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
concept equivalence (definition)	$C \equiv D$	$C^{\mathcal{I}} = D^{\mathcal{I}}$
concept disjointness	$\text{DISJ}(C, D)$	$C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$

Table 2.3: *RBox axiom types.*

Name	Notation	Semantics
atomic role inclusion (hierarchy)	$r \sqsubseteq s$	$r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$
complex role inclusion	$r_1 \circ \dots \circ r_k \sqsubseteq s$	$\left\{ \langle x_0, x_k \rangle \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} : \right.$ there are x_1, x_2, \dots, x_{k-1} in $\Delta^{\mathcal{I}}$ s.t. $\langle x_{i-1}, x_i \rangle \in r_i^{\mathcal{I}}$ for every $i \in \{1, 2, \dots, k\} \left. \right\} \subseteq s^{\mathcal{I}}$
role disjointness	$\text{DISJ}(r, s)$	$r^{\mathcal{I}} \cap s^{\mathcal{I}} = \emptyset$
role transitivity	$\text{TRANS}(r)$	$r^{\mathcal{I}} = \bigcup_{i \geq 0} (r^{\mathcal{I}})^i$
role symmetry	$\text{SYM}(r)$	$\langle y, x \rangle \in r^{\mathcal{I}}$ for all $\langle x, y \rangle \in r^{\mathcal{I}}$
role antisymmetry	$\text{ASYM}(r)$	$\langle y, x \rangle \notin r^{\mathcal{I}}$ for all $\langle x, y \rangle \in r^{\mathcal{I}}$
role reflexivity	$\text{REF}(r)$	$\langle x, x \rangle \in r^{\mathcal{I}}$ for every x in the domain of r
role irreflexivity	$\text{IRREF}(r)$	$\langle x, x \rangle \notin r^{\mathcal{I}}$ for every $x \in \Delta^{\mathcal{I}}$
role functionality	$\text{FUNC}(r)$	$x_1 = x_2$ for all x_0, x_1, x_2 in $\Delta^{\mathcal{I}}$ s.t. $\{\langle x_0, x_1 \rangle, \langle x_0, x_2 \rangle\} \subseteq r^{\mathcal{I}}$

2.1.2 Some usual Description Logics

We will now present a Description Logic called \mathcal{AL} , which is considered to be ‘the minimal sensible attributive concept Description Logic’ [SSS91].

Table 2.4 shows the concept constructors in \mathcal{AL} . There are no complex roles and no RBox axioms in this logic.

Name	Notation	Interpretation
atomic concept	A	$A^{\mathcal{I}}$
universal concept	\top	$\Delta^{\mathcal{I}}$
empty concept	\perp	\emptyset
atomic concept negation	$\neg A$	$\Delta^{\mathcal{I}} \setminus A^{\mathcal{I}}$
intersection (conjunction)	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
value restriction	$\forall r.C$	$\{x \in \Delta^{\mathcal{I}} : \text{for every } y \in \Delta^{\mathcal{I}}, \text{ if } \langle x, y \rangle \in r^{\mathcal{I}}, \text{ then } y \in C^{\mathcal{I}}\}$
limited existential quantification	$\exists r.\top$	$\{x \in \Delta^{\mathcal{I}} : \text{there is some } y \in \Delta^{\mathcal{I}} \text{ such that } \langle x, y \rangle \in r^{\mathcal{I}}\}$

Table 2.4: Concept constructors in \mathcal{AL} .

The logic \mathcal{FL}^- is the sublanguage of \mathcal{AL} obtained by removing the $\neg A$ constructor, and \mathcal{FL} is generated from \mathcal{FL}^- by allowing role restriction (see Table 2.5) [BL84]. \mathcal{FL}_0 can be obtained from \mathcal{FL}^- by the removal of limited existential quantification [Baa96]. The logic \mathcal{EL} allows only the following concept constructors: \top , $\exists r.C$ and $C \sqcap D$ [BKM99].

More expressive logics can be generated from \mathcal{AL} by including some concept and role constructors, such as those listed in Table 2.6 and Table 2.5, and by allowing some types of axioms, such as atomic role inclusion (symbol \mathcal{H}) and role transitivity (symbol \cdot_{R^+}). The usual names of those Description Logics are obtained by appending the symbols of the included features to the name of the original logic (e.g. \mathcal{ALC}). \mathcal{S} is the shortened name of \mathcal{ALC}_{R^+} [BN03, BHLS17].

Symbol	Name	Notation	Interpretation
\mathcal{I}	inverse role	r^-	$\{\langle x, y \rangle \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} : \langle y, x \rangle \in r^{\mathcal{I}}\}$
\cdot_{r^+}	transitive closure	R^+	$\bigcup_{n \geq 1} (r^{\mathcal{I}})^n$
r	role restriction	$r _C$	$\{\langle x, y \rangle \in r^{\mathcal{I}} : y \in C^{\mathcal{I}}\}$

Table 2.5: Some role constructors.

Symbol	Name	Notation	Interpretation
\mathcal{N}	number restriction	$\leq n r$	$\{a \in \Delta^{\mathcal{I}} : \{b \in \Delta^{\mathcal{I}} : \langle a, b \rangle \in r^{\mathcal{I}}\} \leq n\}$
		$\geq n r$	$\{a \in \Delta^{\mathcal{I}} : \{b \in \Delta^{\mathcal{I}} : \langle a, b \rangle \in r^{\mathcal{I}}\} \geq n\}$
\mathcal{C}	complex concept negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
\mathcal{U}	concept union (disjunction) ^(**)	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
\mathcal{E}	existential restriction ^(***)	$\exists r.C$	$\{a \in \Delta^{\mathcal{I}} : \text{there is some } b \in C^{\mathcal{I}} \text{ s.t. } \langle a, b \rangle \in r^{\mathcal{I}}\}$
\mathcal{Q}	qualified number restriction	$\leq n r.C$	$\{a \in \Delta^{\mathcal{I}} : \{b \in C^{\mathcal{I}} : \langle a, b \rangle \in r^{\mathcal{I}}\} \leq n\}$
		$\geq n r.C$	$\{a \in \Delta^{\mathcal{I}} : \{b \in C^{\mathcal{I}} : \langle a, b \rangle \in r^{\mathcal{I}}\} \geq n\}$
\mathcal{O}	nominal	$\{a\}$	$\{a^{\mathcal{I}}\}$

^(**) equivalent to $\neg(\neg C \sqcap \neg D)$, if allowed by the logic

^(***) equivalent to $\neg(\forall r. \neg C)$, if allowed by the logic

Table 2.6: *Some concept constructors.*

2.1.3 OWL

The Web Ontology Language (OWL) is a language that can be used to represent ontologies in a way that applications can automatically process them. Since 2004, W3C¹ has recommended OWL to represent ontologies [MvH04].

As shown by [BL84], there is a trade-off between expressiveness and computational tractability. Table 2.7 shows the diverse set of OWL and OWL 2 profiles (sublanguages), which have been defined to allow flexibility for different types of applications [MvH04, MGH⁺12].

Table 2.7: OWL and OWL 2 profiles.

Profile	Corresponding description logic	Short description
OWL Full	Not a DL	Maximum expressiveness; undecidable
OWL DL	<i>SHOIN</i>	Maximum expressiveness while retaining computational completeness and decidability
OWL Lite	<i>SHIF</i>	Useful for classification hierarchy and simple constraints
OWL 2 Full	Not a DL	Maximum expressiveness; undecidable
OWL 2 DL	<i>SROIQ</i>	High expressiveness; decidable, but computationally expensive
OWL 2 EL	\mathcal{EL}^{++} (<i>ELRO</i>) [BBL05]	Useful for ontologies with large numbers of properties or classes; polynomial-time basic inference
OWL 2 QL	DL-Lite	Aimed at ontologies with large volumes of instance data; efficient query answering
OWL 2 RL	DLP	Can be used with rule-based reasoning engines; polynomial-time inference

The logic \mathcal{EL}^{++} [BBL05] is widely used to represent biological ontologies, such as Gene Ontology (which aims to provide ‘structured, computable knowledge regarding the functions of genes and gene products’ [ABB⁺00, The19]) and SNOMED CT (a ‘comprehensive, multilingual clinical healthcare terminology’²). Most sentences of the clinical terminology GALEN³ can be represented in \mathcal{EL}^{++} , and the full ontology is in *SHIQ*. Most features of FMA (Foundation Model of Anatomy) can be represented in *SHOIN* [GZB06].

2.2 Belief Revision

Rational agents, be they humans or machines, must have some representation of their knowledge or belief system. *Belief Revision* is a research area that aims at understanding how these agents should change these representations when they are faced with new information.

¹The World Wide Web Consortium (W3C) is ‘an international community that develops open standards to ensure the long-term growth of the Web’. <https://www.w3.org/>

²<https://www.snomed.org/snomed-ct/five-step-briefing>

³<http://www.opengalen.org/>

2.2.1 AGM Framework

Alchourrón, Gärdenfors and Makinson (henceforth, AGM) published the paper that is considered the initial hallmark of the area of Belief Revision [AGM85], where they proposed to represent epistemic states by logically closed sets of propositions (called *belief sets* or *theories*), i.e., a set K such that $K = \text{Cn}(K)$ for some consequence relation Cn (usually Tarskian).

Three change operations on the agent's knowledge K with respect to a sentence α were considered⁴:

- Expansion ($\text{ex}(K, \alpha)$): the sentence α is incorporated to K possibly leading to an inconsistency.
- Revision ($\text{r}(K, \alpha)$): the sentence α is incorporated to K in a way that the resulting set is consistent.
- Contraction ($\text{c}(K, \alpha)$): the sentence α is removed from K and must not be entailed by the contracted set.

Only expansion was explicitly defined: $\text{ex}(K, \alpha) = \text{Cn}(K \cup \{\alpha\})$. Revision and contraction were instead constrained by rationality postulates, which aim to avoid unnecessary additions or removals according to the *principle of minimal change*.

These are the basic AGM postulates for contraction, where α and α' are sentences and K is a theory [AGM85]:

(**closure**) $\text{c}(K, \alpha) = \text{Cn}(\text{c}(K, \alpha))$.

(**inclusion**)⁵ $\text{c}(K, \alpha) \subseteq K$.

(**vacuity**) If $\alpha \notin K$, then $\text{c}(K, \alpha) = K$.

(**success**) If $\alpha \notin \text{Cn}(\emptyset)$, then $\alpha \notin \text{c}(K, \alpha)$.

(**extensionality (or preservation)**) If $\text{Cn}(\{\alpha\}) = \text{Cn}(\{\alpha'\})$, then $\text{c}(K, \alpha) = \text{c}(K, \alpha')$.

(**recovery**) $K \subseteq \text{ex}(\text{c}(K, \alpha), \alpha)$.

Revision also has its set of AGM postulates [Gär88]:

(**closure**) $\text{r}(K, \alpha) = \text{Cn}(\text{r}(K, \alpha))$.

(**success**) $\alpha \in \text{r}(K, \alpha)$.

(**inclusion**) $\text{r}(K, \alpha) \subseteq \text{Cn}(K \cup \{\alpha\})$.

(**vacuity**) If $\perp \notin \text{Cn}(K \cup \{\alpha\})$, then $\text{r}(K, \alpha) \supseteq \text{Cn}(K \cup \{\alpha\})$ ⁶.

(**consistency**) $\perp \notin \text{Cn}(\text{r}(K, \alpha))$ unless $\perp \in \text{Cn}(\{\alpha\})$.

(**extensionality**) If $\text{Cn}(\{\alpha\}) = \text{Cn}(\{\alpha'\})$, then $\text{r}(K, \alpha) = \text{r}(K, \alpha')$.

In [AGM85], a construction for a contraction operation known as *partial meet contraction* was proposed, based on the idea of selecting maximal subsets of the agent's knowledge that do not entail the sentence being contracted.

⁴The classical notation for expansion, revision and contraction is $K + \alpha$, $K * \alpha$ and $K - \alpha$, respectively. The remainder set and the kernel set (defined ahead) were represented respectively by $K \perp \alpha$ and $K \perp\!\!\!\perp \alpha$. We changed the notation to avoid using the symbol \perp for multiple purposes, to make it clear that the operations are functions and to facilitate the explicit representation of specific types of operations.

⁵This is unrelated to the inclusion property of consequence operators.

⁶Vacuity and inclusion imply that $\text{r}(K, \alpha) = \text{Cn}(K \cup \{\alpha\})$ if $K \cup \{\alpha\}$ is consistent.

Definition 2.1 (Remainder and remainder set [AGM85]). Let $K \subseteq \mathfrak{L}$ and $\alpha \in \mathfrak{L}$. The *remainder set* of K with respect to α , denoted by $\text{Rem}[K, \alpha]$, is the set of all $X \subseteq K$ such that $\alpha \notin \text{Cn}(X)$ and there is no Y such that $X \subset Y \subseteq K$ and $\alpha \notin \text{Cn}(Y)$. Each such X is an α -*remainder* of K .

In order to compute the contraction, at least one of the α -remainders is selected, according to some preference criteria, encoded as a function:

Definition 2.2 (Selection function [AGM85]). Let $K \subseteq \mathfrak{L}$. A function \mathbf{g} is a *selection function* for K if, for every $\alpha \in \mathfrak{L}$, it is the case that $\emptyset \neq \mathbf{g}(\text{Rem}[K, \alpha]) \subseteq \text{Rem}[K, \alpha]$ if $\text{Rem}[K, \alpha]$ is non-empty, or $\mathbf{g}(\text{Rem}[K, \alpha]) = \{K\}$ otherwise.

The sentences in the selected α -remainders form the resulting contracted set:

Definition 2.3 (Partial meet contraction [AGM85]). Let $K \subseteq \mathfrak{L}$, and let \mathbf{g} be a selection function for K . The *partial meet contraction* of K by a sentence α , denoted by $\text{pmc}_{\mathbf{g}}(K, \alpha)$, is defined as $\bigcap \mathbf{g}(\text{Rem}[K, \alpha])$.

It turns out that partial meet contraction is exactly equivalent to the aforementioned postulates for contraction, giving us one of the most important results in Belief Revision:

Theorem 2.4 (Partial meet contraction for theories: representation theorem [AGM85]). *An operation c is a partial meet contraction for a theory K if and only if c satisfies the six basic AGM postulates for contraction.*

Revision is usually defined in terms of a contraction c by Levi identity:

Definition 2.5 (Levi identity [AGM85]). Let c be a contraction operation, $K \subseteq \mathfrak{L}$ be a theory, and $\alpha \in \mathfrak{L}$. The c -*based revision* of K by α , denoted by $r_{\text{Levi}(c)}(K, \alpha)$, is defined as $\text{ex}(c(K, \neg\alpha), \alpha)$, i.e., $r_{\text{Levi}(c)}(K, \alpha) = \text{Cn}(c(K, \neg\alpha) \cup \{\alpha\})$.

If c satisfies the AGM postulates for contraction (recovery is not needed), then $r_{\text{Levi}(c)}$ satisfies the postulates for revision:

Theorem 2.6 (Postulates satisfied by the theory revision obtained by Levi identity [Gär88]). *If c is a contraction operation that satisfies closure, inclusion, vacuity, success and extensionality, then $r_{\text{Levi}(c)}$ satisfies closure, success, inclusion, vacuity, consistency and extensionality.*

Conversely, we can obtain a contraction from a revision operation⁷:

Definition 2.7 (Harper identity [Gär88]). Let r be a revision operation, $K \subseteq \mathfrak{L}$ and $\alpha \in \mathfrak{L}$. The r -*based contraction* of K by α , denoted by $c_{\text{Harper}(r)}(K, \alpha)$, is defined as $c_{\text{Harper}(r)}(K, \alpha) = K \cap r(K, \neg\alpha)$.

The contraction operation obtained by Harper identity satisfies the postulates for contraction.

Theorem 2.8 (Postulates satisfied by the theory contraction obtained by Harper identity [Gär88]). *If r is a revision operation that satisfies the six AGM postulates for revision, then $c_{\text{Harper}(r)}$ satisfies the six AGM postulates for contraction.*

If we start with a contraction operation, we can obtain a revision construction using Levi identity, from which we can produce another contraction using Harper identity. This contraction ought to be identical to the original one. The same happens if we start with a revision and use Harper and Levi identities.

Theorem 2.9 (Cycle obtained by Levi and Harper identities [Gär88]). *For any revision r and contraction c , $r_{\text{Levi}(c_{\text{Harper}(r)})}$ is identical to r , and $c_{\text{Harper}(r_{\text{Levi}(c)})}$ is identical to c .*

⁷This operation is also called *internal revision*. There is also an *external revision*, which will not be covered by this text.

2.2.2 Generalisation for belief bases

Belief bases are arbitrary, not necessarily closed, sets of sentences, which grants them greater expressivity when compared to belief sets. The AGM theory was generalised for belief sets by [Han93b, Han99].

Expansion is defined as $\text{ex}(B, \alpha) = B \cup \{\alpha\}$, which does not close the resulting set [Han99].

For partial meet contraction, the same construction of Definition 2.3 can be applied to belief bases:

Definition 2.10 (Partial meet contraction for bases [Han93b]). Let $B \subseteq \mathcal{L}$ be a belief base, and let \mathbf{g} be a selection function for B . The *partial meet contraction* of B by a sentence α , denoted by $\text{pmc}_{\mathbf{g}}(B, \alpha)$, is defined as $\bigcap \mathbf{g}(\text{Rem}[B, \alpha])$.

Despite having the same definition, not all AGM postulates are satisfied by this operation. In particular, the *recovery* postulate is not satisfied by partial meet base contraction. Hansson has characterised this operation for bases by means of the following postulates:

(success) If $\alpha \notin \text{Cn}(\emptyset)$, then $\alpha \notin \text{Cn}(c(B, \alpha))$.

(inclusion) $c(B, \alpha) \subseteq B$.

(relevance) If $\beta \in B \setminus c(B, \alpha)$ ⁸, then there is a B' such that $c(B, \alpha) \subseteq B' \subseteq B$ and $\alpha \in \text{Cn}(B' \cup \{\beta\}) \setminus \text{Cn}(B')$.

(uniformity) If for all $B' \subseteq B$, $\alpha \in \text{Cn}(B')$ if and only if $\beta \in \text{Cn}(B')$, then $c(B, \alpha) = c(B, \beta)$.

Theorem 2.11 (Partial meet base contraction: representation theorem [Han93b]). *An operation c is a partial meet contraction for a belief base B if and only if for all sentences α the operation $c(B, \alpha)$ satisfies success, inclusion, relevance and uniformity.*

The following useful properties will be needed later in this text:

Proposition 2.12 (Upper Bound Property [AM81]). *If $X \subseteq B$ and $\alpha \notin \text{Cn}(X)$, then there is some X' such that $X \subseteq X'$ and $X' \in \text{Rem}[B, \alpha]$.*

Proposition 2.13 (Necessary and sufficient condition for equality of remainder sets of the same base [Han99]). *Let B be a set of sentences. For all sentences α and α' , the following two conditions are equivalent:*

- $\text{Rem}[B, \alpha] = \text{Rem}[B, \alpha']$;
- for all $X \subseteq B$, $\alpha \in \text{Cn}(X)$ if and only if $\alpha' \in \text{Cn}(X)$.

Another construction for contraction was proposed in [Han94], which relies on minimal sets implying the undesirable sentence:

Definition 2.14 (Kernel and kernel set [Han94]). Let $B \subseteq \mathcal{L}$ and $\alpha \in \mathcal{L}$. The *kernel set* of B with respect to α , denoted by $\text{Ker}[B, \alpha]$, is such that a set X is in $\text{Ker}[B, \alpha]$ if and only if $X \subseteq B$, $\alpha \in \text{Cn}(X)$, and there is no $Y \subset X$ such that $\alpha \in \text{Cn}(Y)$. Each such X is an α -kernel.

In order to contract α from B , at least one element of each α -kernel must be removed.

Definition 2.15 (Incision function [Han94]). Let $B \subseteq \mathcal{L}$. A function \mathbf{f} is an *incision function* for B if, for every $\alpha \in \mathcal{L}$, it is the case that $\mathbf{f}(\text{Ker}[B, \alpha]) \subseteq \bigcup \text{Ker}[B, \alpha]$ and $\mathbf{f}(\text{Ker}[B, \alpha]) \cap X \neq \emptyset$ for every non-empty $X \in \text{Ker}[B, \alpha]$.

⁸The symbol \setminus denotes set difference: $X \setminus Y = \{x : x \in X \text{ and } x \notin Y\}$.

The incision function chooses a subset of the kernel set that contains at least one sentence of each α -kernel. The contracted set is obtained by removing from the original belief base the elements of this subset:

Definition 2.16 (Kernel contraction [Han94]). Let $B \subseteq \mathcal{L}$, and let \mathbf{f} be an incision function for B . The *kernel contraction* of B by a sentence α , denoted by $\text{kc}_{\mathbf{f}}(B, \alpha)$, is defined as $B \setminus \mathbf{f}(\text{Ker}[B, \alpha])$.

The following example⁹ illustrates this operation.

Example 2.17. Let

$$B = \{\text{alphaZero} : \text{ChessPlayer}, \text{alphaZero} : \text{GoPlayer}, \text{ChessPlayer} \sqcup \text{GoPlayer} \sqsubseteq \text{Person}, \text{alphaZero} : \text{ComputerProgram}\},$$

and let α be the sentence $\text{alphaZero} : \text{Person}$. The kernel set of B with respect to α is $\text{Ker}[B, \alpha] = \{\{\text{alphaZero} : \text{ChessPlayer}, \text{ChessPlayer} \sqcup \text{GoPlayer} \sqsubseteq \text{Person}\}, \{\text{alphaZero} : \text{GoPlayer}, \text{ChessPlayer} \sqcup \text{GoPlayer} \sqsubseteq \text{Person}\}\}$. An incision function \mathbf{f} may choose the following set of sentences to remove, because it contains at least one element from each α -kernel:

$$\mathbf{f}(\text{Ker}[B, \alpha]) = \{\text{alphaZero} : \text{GoPlayer}, \text{ChessPlayer} \sqcup \text{GoPlayer} \sqsubseteq \text{Person}\}.$$

The resulting set is $\text{kc}_{\mathbf{f}}(B, \alpha) = \{\text{alphaZero} : \text{ChessPlayer}, \text{alphaZero} : \text{ComputerProgram}\}$.

Example 2.17 shows that kernel contraction does not necessarily satisfy relevance: the sentence $\text{alphaZero} : \text{GoPlayer}$ is in B and not in $\text{kc}_{\mathbf{f}}(B, \alpha)$, but there is no set B' such that $\{\text{alphaZero} : \text{ChessPlayer}\} \subseteq B' \subseteq B$ and $\alpha \in \text{Cn}(B' \cup \{\text{alphaZero} : \text{GoPlayer}\}) \setminus \text{Cn}(B')$. Intuitively, relevance would not allow the unnecessary removal of $\text{alphaZero} : \text{GoPlayer}$.

Instead of relevance, kernel contraction satisfies a slightly more general property:

(core-retainment) If $\beta \in B \setminus c(B, \alpha)$, then there is some B' such that $B' \subseteq B$ and $\alpha \in \text{Cn}(B' \cup \{\beta\}) \setminus \text{Cn}(B')$.

Kernel contraction can also be characterised by a set of rationality postulates:

Theorem 2.18 (Kernel contraction: representation theorem [Han94]). *An operation c is a kernel contraction for a belief base B if and only if c satisfies inclusion, success, uniformity and core-retainment.*

From this theorem, as core-retainment is more general than relevance, we see that all partial meet contractions can be constructed as a kernel contraction [Han94], but the opposite does not hold, i.e., kernel contractions are more general than partial meet contractions.

A desirable property that partial meet contraction satisfies but kernel contraction does not is *relative closure* [Han91]:

(relative closure) $B \cap \text{Cn}(c(B, \alpha)) \subseteq c(B, \alpha)$.

This property is a consequence of the postulate of relevance [Han99], which kernel contraction does not satisfy. Nevertheless, a special type of kernel contraction called *smooth kernel contraction* [Han94] satisfies relative closure:

Definition 2.19 (Smooth incision function and smooth kernel contraction [Han94]). An incision function \mathbf{f} for a set B is *smooth* if and only if $B' \cap \mathbf{f}(\text{Ker}[B, \alpha]) \neq \emptyset$ for all $B' \subseteq B$ such that $\text{Cn}(B') \cap \mathbf{f}(\text{Ker}[B, \alpha]) \neq \emptyset$. A kernel contraction $\text{kc}_{\mathbf{f}}$ is *smooth* if and only if \mathbf{f} is smooth.

⁹Classical Belief Revision was developed having propositional logic in mind. We use Description Logic knowledge bases in our examples because this text is focused on that type of logic, but the examples could be easily adapted to propositional or first-order logic.

For belief bases, Levi identity is slightly different from the identity for theories.

Definition 2.20 (Levi identity for bases [Han93b]). Let c be a contraction operation, $B \subseteq \mathfrak{L}$ be a belief base, and $\alpha \in \mathfrak{L}$. The c -based revision of B by α , denoted by $r_{\text{bLevi}(c)}(B, \alpha)$, is defined as $\text{ex}(c(B, \neg\alpha), \alpha)$, i.e., $r_{\text{bLevi}(c)}(B, \alpha) = c(B, \neg\alpha) \cup \{\alpha\}$.

When the Levi identity is applied to a partial meet contraction, the resulting operation is called *internal partial meet revision*¹⁰:

Definition 2.21 (Internal partial meet revision [Han99]). Let $\text{pmc}_{\mathbf{g}}$ be a partial meet contraction for bases, $B \subseteq \mathfrak{L}$ be a belief base, and $\alpha \in \mathfrak{L}$. The *internal partial meet revision of B by α* , denoted by $\text{pmr}_{\mathbf{g}}(B, \alpha)$, is defined as $\text{ex}(\text{pmc}_{\mathbf{g}}(B, \neg\alpha), \alpha)$, i.e., $\text{pmr}_{\mathbf{g}}(B, \alpha) = \bigcap (\mathbf{g}(\text{Rem}[B, \neg\alpha])) \cup \{\alpha\}$.

This generalisation of revision for belief bases has been characterised by means of the following postulates [Han99]:

(**success**) $\alpha \in r(B, \alpha)$.

(**consistency**) $\perp \notin \text{Cn}(r(B, \alpha))$, unless $\perp \in \text{Cn}(\{\alpha\})$.

(**inclusion**) $r(B, \alpha) \subseteq B \cup \alpha$.

(**relevance**) If $\beta \in B \setminus r(B, \alpha)$, then there is some B' such that $r(B, \alpha) \subseteq B' \subseteq B \cup \alpha$ and $\perp \in \text{Cn}(B' \cup \{\beta\}) \setminus \text{Cn}(B')$.

(**uniformity**) If for all $B' \subseteq B$, $\perp \in \text{Cn}(B' \cup \{\alpha\})$ if and only if $\perp \in \text{Cn}(B' \cup \{\beta\})$, then $B \cap r(B, \alpha) = B \cap r(B, \beta)$.

Theorem 2.22 (Partial meet revision: representation theorem [Han99]). *The operation r is an internal partial meet revision if and only if r satisfies success, consistency, inclusion, relevance and uniformity.*

It is important to note that Levi identity cannot be (directly) applied if the underlying logic is not closed under negation (i.e., if $\{\neg\alpha : \alpha \in \mathfrak{L}\} \setminus \mathfrak{L} \neq \emptyset$). For base revision in those logics, see [Rib13].

In this text, we will focus on contractions, and its generalisations called *pseudo-contractions*, which will be introduced in the next chapter.

¹⁰There is also an *external partial meet revision* [Han99], defined as $\text{pmc}_{\mathbf{g}}(\text{ex}(B, \alpha), \neg\alpha)$, but we will not discuss it here.

Chapter 3

Pseudo-contraction Operations

Partial meet and kernel contractions over belief bases can lead to unnecessary waste of information, largely due to the inclusion postulate. This postulate prevents the weakening of formulae, which can be seen as an argument against its use for belief bases.

Example 3.1. Consider a knowledge base that contains the following sentences:

$$\begin{aligned} & \textit{Swedish} \sqsubseteq \textit{European}; \\ & \textit{European} \sqcap \textit{Swan} \sqsubseteq \exists \textit{hasMainColour}.\{\textit{white}\}; \\ & s : \textit{Swan} \sqcap \textit{Swedish}. \end{aligned}$$

If we want to contract by $s \textit{ hasMainColour white}$, one of the three sentences must be removed; thus, for example, if we choose to remove the third sentence, the fact that s is a swan is lost.

Intuitively, in [Example 3.1](#), we should consider replacing the sentence $s : \textit{Swan} \sqcap \textit{Swedish}$ with a weaker version $s : \textit{Swan}$, which is forbidden by the inclusion postulate. Another intuitive idea prohibited by that postulate is to weaken $\textit{European} \sqcap \textit{Swan} \sqsubseteq \exists \textit{hasMainColour}.\{\textit{white}\}$ by adding an intersection to the left-hand side, in order to convey the idea that all European swans that satisfy a certain property (e.g. ‘normal’ or ‘typical’) are white.

Hansson has proposed a weakening of inclusion, *logical inclusion* [[Han89](#)], which is satisfied by operations he has called *pseudo-contractions* [[Han93a](#)]:

(logical inclusion) $\text{Cn}(c(B, \alpha)) \subseteq \text{Cn}(B)$.

Definition 3.2 (Pseudo-contraction [[Han93a](#)]). An operation c is a *pseudo-contraction* if c satisfies success and logical inclusion.

With logical inclusion, whilst we still do not allow the addition of arbitrary sentences (as no new consequences can be added), the resulting set no longer has to be a subset of the original set, thus making it possible to insert the sentence $s : \textit{Swan}$ in [Example 3.1](#).

We will also utilise the following postulate, which is similar to relevance (the only difference is the fact that relevance requires a B' that is a subset of B , whereas being a subset of $\text{Cn}(B)$ is enough for logical relevance):

(logical relevance) If $\beta \in B \setminus c(B, \alpha)$, then there is a B' such that $c(B, \alpha) \subseteq B' \subseteq \text{Cn}(B)$ and $\alpha \in \text{Cn}(B' \cup \{\beta\}) \setminus \text{Cn}(B')$.

From now on, we will consider a generic consequence operator Cn that is Tarskian and compact, such as Cn_{FOL} and the consequence operators that correspond to some fragments of first-order logic. Subclassicality will be defined with respect to Cn . The symbol Cn^* will be used to represent ‘weak’ consequence operators.

In the following sections, we will present some pseudo-contraction constructions that depend on the kind of formulae that we are allowed to add when contracting by a formula. Before computing the remainder set or the kernel set, our operations will ‘close’ the set under a new consequence operator, Cn^* , which will make possible the insertion of sentences that were not in the original belief base. The properties of pseudo-contraction constructions depend on the properties that are satisfied by Cn^* , especially inclusion and subclassicality: if both are satisfied, then Cn^* is in an intermediate level between the original base (which would be used in a base contraction) and its closure (as in classical AGM contraction), i.e. $B \subseteq Cn^*(B) \subseteq Cn(B)$. For practical applications, $Cn^*(B)$ should always be finite if B is finite, but we will not assume this restriction.

Automatic reasoners for ontologies — such as Hermit¹ and Fact++² — allow the user to choose the types of consequences that will be generated. Each configuration can be seen as a Cn^* , which satisfies inclusion³ and is usually subclassical⁴. Those reasoners can be embedded in ontology editors, such as Protégé⁵, as shown in Figure 3.1.

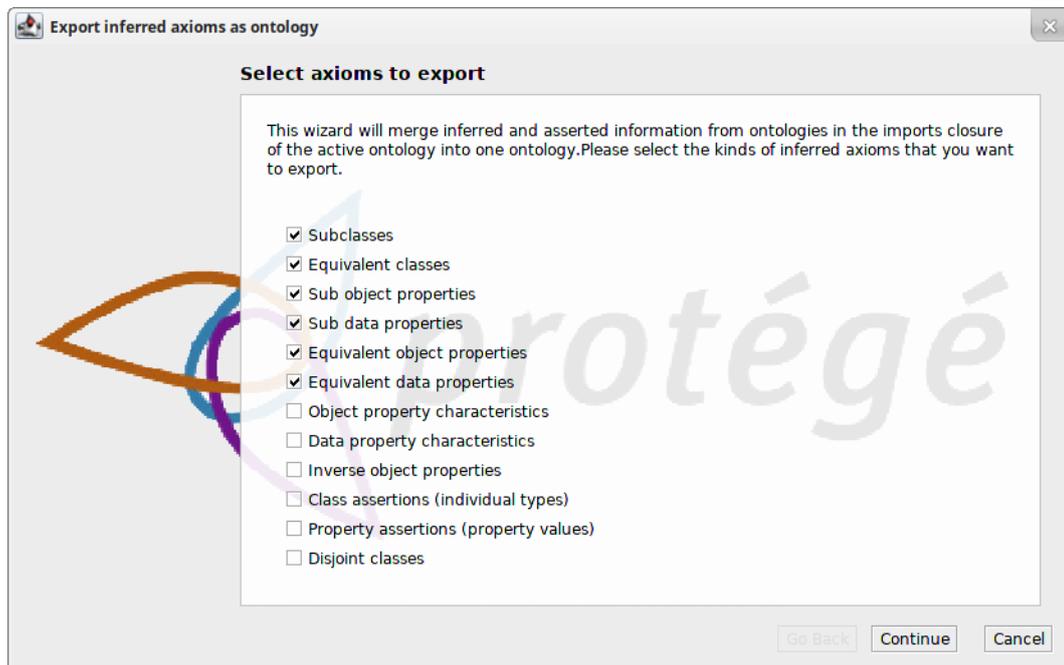


Figure 3.1: Screenshot of the Protégé window where the user can choose which types of consequences will be exported.

Another example of Cn^* is the conversion of sentences into some standard format, such as conjunctive normal form, disjunctive normal form, or normal forms tailored to specific logics, e.g. normal form for \mathcal{EL}^{++} [BBL05]. Since inclusion is not satisfied by those consequence operators, some desirable properties may not be satisfied by constructions that use them; on the other hand, they may be of use if good inference algorithms exist for them, such as [DM00, BBL05].

3.1 Cn^* partial meet pseudo-contraction

This section describes and characterises a pseudo-contraction operation that is based on partial meet contraction. It works in a similar fashion as partial meet contraction for belief bases; however, before computing the remainder set, the Cn^* partial meet pseudo-contraction ‘closes’ the set under

¹<http://www.hermit-reasoner.com/>

²<http://owl.cs.manchester.ac.uk/tools/fact/>

³One step after the window shown in Figure 3.1, users can choose whether the original sentences should be kept.

⁴Except, maybe, for some highly-complex ontologies that cannot be represented in classical logic.

⁵<http://protege.stanford.edu/>

a consequence operator Cn^* , which will allow the insertion of sentences into the belief base during the pseudo-contraction.

The ideas and results of this section were first presented in Santos's thesis [San16] and published in [SMRW18]⁶. We first introduce the operation, and then show its formal properties.

Definition 3.3 (Cn^* partial meet pseudo-contraction [San16]). Let B be a set of sentences, Cn^* a consequence relation and \mathbf{g} a selection function for $\text{Cn}^*(B)$. The Cn^* partial meet pseudo-contraction of B by a sentence α , denoted by $\text{pmc}_{\mathbf{g}}^{\text{Cn}^*}(B, \alpha)$, is such that, for all sentences α :

$$\text{pmc}_{\mathbf{g}}^{\text{Cn}^*}(B, \alpha) = \bigcap \mathbf{g}(\text{Rem}[\text{Cn}^*(B), \alpha]).$$

To illustrate this operation, we will revisit the example that introduced this chapter.

Example 3.4. Consider again the problem we discussed in Example 3.1. Let $\text{Cn}^*_{\text{break}}$ be a consequence operator that preserves the existing sentences and adds the sentences $a : C_i$ (for $i = 1, \dots, n$) for every sentence $a : C_1 \sqcap \dots \sqcap C_n$ in the original set⁷. If B is the knowledge base defined in the aforementioned example, then $\text{Cn}^*_{\text{break}}(B) = B \cup \{s : \text{Swan}, s : \text{Swedish}\}$. The set obtained by replacing $s : \text{Swan} \sqcap \text{Swedish}$ with $s : \text{Swan}$ (which was our desired output) is in $\text{Rem}[\text{Cn}^*_{\text{break}}(B), \alpha]$, which makes it a valid Cn^* partial meet pseudo-contraction (as \mathbf{g} may select that set).

We will now give a characterisation for this operation, but before that we need to introduce the following 'starred' version of some postulates.

(inclusion*) $c(B, \alpha) \subseteq \text{Cn}^*(B)$.

(relevance*) If $\beta \in \text{Cn}^*(B) \setminus c(B, \alpha)$, then there is a B' such that $c(B, \alpha) \subseteq B' \subseteq \text{Cn}^*(B)$ and $\alpha \in \text{Cn}(B' \cup \{\beta\}) \setminus \text{Cn}(B')$.

(uniformity*) If for all $B' \subseteq \text{Cn}^*(B)$, $\alpha \in \text{Cn}(B')$ if and only if $\beta \in \text{Cn}(B')$, then $c(B, \alpha) = c(B, \beta)$.

It is not surprising that, in order to characterise this operation, which embeds two different logics, we would need to mention this second logic Cn^* in the postulates. The representation theorem is the following:

Theorem 3.5 (Cn^* partial meet pseudo-contraction: representation theorem [San16]). *Provided that Cn^* satisfies inclusion, idempotence and subclassicality, an operation is a Cn^* partial meet pseudo-contraction if and only if it satisfies success, inclusion*, relevance* and uniformity*.*

*Proof.*⁸

Construction-to-postulates: We know from the definitions of $\text{pmc}_{\mathbf{g}}^{\text{Cn}^*}$ and $\text{pmc}_{\mathbf{g}}$ that $\text{pmc}_{\mathbf{g}}^{\text{Cn}^*}(B, \alpha) = \bigcap \mathbf{g}(\text{Rem}[\text{Cn}^*(B), \alpha]) = \text{pmc}_{\mathbf{g}}(\text{Cn}^*(B), \alpha)$. We also know that $\text{pmc}_{\mathbf{g}}$ satisfies success, inclusion, relevance and uniformity. We have:

- If $\alpha \notin \text{Cn}(\emptyset)$, then $\alpha \notin \text{Cn}(\text{pmc}_{\mathbf{g}}(\text{Cn}^*(B), \alpha))$;
- $\text{pmc}_{\mathbf{g}}(\text{Cn}^*(B), \alpha) \subseteq \text{Cn}^*(B)$;
- If $\beta \in \text{Cn}^*(B) \setminus \text{pmc}_{\mathbf{g}}(\text{Cn}^*(B), \alpha)$, then there is a B' such that $\text{pmc}_{\mathbf{g}}(\text{Cn}^*(B), \alpha) \subseteq B' \subseteq \text{Cn}^*(B)$ and $\alpha \in \text{Cn}(B' \cup \{\beta\}) \setminus \text{Cn}(B')$;

⁶ In [SMRW18], the operation was denoted by $-_*$ and was called Cn^* -pseudo-contraction. In this text, we added 'partial meet' to its name because we shall also discuss a similar construction that is based on kernel contraction.

⁷This is analogous to the consequence operator that 'breaks conjunctions into conjuncts' presented in [San16].

⁸Although this proof is not new (it was presented in [San16]), we did not omit it here, because it is the most important result about Cn^* partial meet pseudo-contractions; moreover, it might be a good demonstration of the notation that we are proposing.

- If for all $B' \subseteq \text{Cn}^*(B)$, $\alpha \in \text{Cn}(B')$ if and only if $\beta \in \text{Cn}(B')$, then $\text{pmc}_{\mathbf{g}}(\text{Cn}^*(B), \alpha) = \text{pmc}_{\mathbf{g}}(\text{Cn}^*(B), \beta)$.

Since $\text{pmc}_{\mathbf{g}}(\text{Cn}^*(B), \alpha) = \text{pmc}_{\mathbf{g}}^{\text{Cn}^*}(B, \alpha)$, we are done.

Postulates-to-construction: This part is almost trivially obtained from the proof of the representation theorem for partial meet contraction for bases, which can be found in [Han93b].

Let c^{Cn^*} be an operation for B that satisfies success, inclusion*, relevance* and uniformity*. Let \mathbf{g} be a function such that:

- If $\text{Rem}[\text{Cn}^*(B), \alpha] = \emptyset$, then $\mathbf{g}(\text{Rem}[\text{Cn}^*(B), \alpha]) = \{\text{Cn}^*(B)\}$;
- Otherwise, $\mathbf{g}(\text{Rem}[\text{Cn}^*(B), \alpha]) = \{X \in \text{Rem}[\text{Cn}^*(B), \alpha] : c^{\text{Cn}^*}(B, \alpha) \subseteq X\}$.

We need to show that (1) \mathbf{g} is a well-defined function, (2) \mathbf{g} is a selection function and (3) $\text{pmc}_{\mathbf{g}}^{\text{Cn}^*}(B, \alpha) = c^{\text{Cn}^*}(B, \alpha)$ for all α .

- (1) For \mathbf{g} to be a well-defined function, for all sentences α_1 and α_2 , if $\text{Rem}[\text{Cn}^*(B), \alpha_1] = \text{Rem}[\text{Cn}^*(B), \alpha_2]$, we must have $\bigcap \mathbf{g}(\text{Rem}[\text{Cn}^*(B), \alpha_1]) = \bigcap \mathbf{g}(\text{Rem}[\text{Cn}^*(B), \alpha_2])$. Suppose that B is such that the set of its α_1 -remainders and the set of its α_2 -remainders are the same, i.e., $\text{Rem}[\text{Cn}^*(B), \alpha_1] = \text{Rem}[\text{Cn}^*(B), \alpha_2]$. From Proposition 2.13, it follows that any subset of $\text{Cn}^*(B)$ implies α_1 if and only if it implies α_2 . By uniformity*, $c^{\text{Cn}^*}(\text{Cn}^*(B), \alpha_1)$ must be equal to $c^{\text{Cn}^*}(\text{Cn}^*(B), \alpha_2)$. By the definition of \mathbf{g} we have that $\mathbf{g}(\text{Rem}[\text{Cn}^*(\text{Cn}^*(B)), \alpha_1]) = \mathbf{g}(\text{Rem}[\text{Cn}^*(\text{Cn}^*(B)), \alpha_2])$. Since Cn^* satisfies idempotence, the result follows.
- (2) For \mathbf{g} to be a selection function, it remains to be proved that $\mathbf{g}(\text{Rem}[\text{Cn}^*(B), \alpha]) \neq \emptyset$ unless $\text{Rem}[\text{Cn}^*(B), \alpha]$ is empty. Then, assuming $\text{Rem}[\text{Cn}^*(B), \alpha] \neq \emptyset$, we know that there is at least one $X \in \text{Rem}[\text{Cn}^*(B), \alpha]$, and we must show that at least one of these X contains $c^{\text{Cn}^*}(B, \alpha)$. Since $\text{Rem}[\text{Cn}^*(B), \alpha]$ is not empty, it must be the case that $\alpha \notin \text{Cn}(\emptyset)$, and by success, $\alpha \notin \text{Cn}(c^{\text{Cn}^*}(B, \alpha))$. By inclusion*, $c^{\text{Cn}^*}(B, \alpha) \subseteq \text{Cn}^*(B)$; then, by the upper bound property (Proposition 2.12), there is some $B' \in \text{Rem}[\text{Cn}^*(B), \alpha]$ such that $c^{\text{Cn}^*}(B, \alpha) \subseteq B'$. By the construction of \mathbf{g} , we have that $\mathbf{g}(\text{Rem}[\text{Cn}^*(B), \alpha])$ is non-empty.
- (3)
 - Case 1: $\alpha \in \text{Cn}(\emptyset)$. As c^{Cn^*} satisfies relevance and Cn^* is assumed to satisfy inclusion and subclassicality, from Observation 3.7 (ahead) we have that c^{Cn^*} satisfies logical relevance; then, since there is no B' such that $\alpha \notin \text{Cn}(B')$, no element is in $B \setminus c^{\text{Cn}^*}(B, \alpha)$, and by inclusion* we have $B \subseteq c^{\text{Cn}^*}(B, \alpha) \subseteq \text{Cn}^*(B)$. Since $\text{Rem}[\text{Cn}^*(B), \alpha] = \emptyset$, by definition we know that $\text{pmc}_{\mathbf{g}}^{\text{Cn}^*}(B, \alpha) = \bigcap \mathbf{g}(\text{Rem}[\text{Cn}^*(B), \alpha]) = \text{Cn}^*(B)$, so we just have to show that $\text{Cn}^*(B) = c^{\text{Cn}^*}(B, \alpha)$. Inclusion* gives $\text{Cn}^*(B) \supseteq c^{\text{Cn}^*}(B, \alpha)$, and the fact that $\text{Cn}^*(B) \subseteq c^{\text{Cn}^*}(B, \alpha)$ is a consequence of relevance* (otherwise, we would have that $\text{Cn}^*(B) \setminus c^{\text{Cn}^*}(B, \alpha) \neq \emptyset$, and relevance* would imply the existence of a set B' that does not entail α , which is impossible because $\alpha \in \text{Cn}(\emptyset)$ and Cn is monotonic).
 - Case 2: $\alpha \notin \text{Cn}(\emptyset)$. The set $\text{Rem}[\text{Cn}^*(B), \alpha]$ is non-empty and (by part 2) so is $\mathbf{g}(\text{Rem}[\text{Cn}^*(B), \alpha])$. Since $c^{\text{Cn}^*}(B, \alpha)$ is a subset of all elements of $\mathbf{g}(\text{Rem}[\text{Cn}^*(B), \alpha])$, $c^{\text{Cn}^*}(B, \alpha) \subseteq \bigcap \mathbf{g}(\text{Rem}[\text{Cn}^*(B), \alpha])$. We need to show that $\bigcap \mathbf{g}(\text{Rem}[\text{Cn}^*(B), \alpha]) \subseteq c^{\text{Cn}^*}(B, \alpha)$. Take $\varepsilon \notin c^{\text{Cn}^*}(B, \alpha)$. If $\varepsilon \notin \text{Cn}^*(B)$, then obviously $\varepsilon \notin \bigcap \gamma(\text{Rem}[\text{Cn}^*(B), \alpha])$, so we just have to consider the case where $\varepsilon \in \text{Cn}^*(B) \setminus c^{\text{Cn}^*}(B, \alpha)$. By relevance*, there is a B' such that $c^{\text{Cn}^*}(B, \alpha) \subseteq B' \subseteq \text{Cn}^*(B)$, $\alpha \notin \text{Cn}(B')$ but $\alpha \in \text{Cn}(B' \cup \{\varepsilon\})$. It follows from the upper bound property (Proposition 2.12) that there is a B'' such that $B' \subseteq B''$ and $B'' \in \text{Rem}[\text{Cn}^*(B), \alpha]$. Since $B' \subseteq B''$ and $\alpha \in \text{Cn}(B' \cup \{\varepsilon\})$, if $\varepsilon \in B''$, from monotonicity of Cn we would conclude that $\alpha \in \text{Cn}(B'')$; hence, we must have $\varepsilon \notin B''$. By our definition of \mathbf{g} , $B'' \in \mathbf{g}(\text{Rem}[\text{Cn}^*(B), \alpha])$, and since $\varepsilon \notin B''$, we conclude that $\varepsilon \notin \bigcap \mathbf{g}(\text{Rem}[\text{Cn}^*(B), \alpha])$.

■

Despite the fact that the non-starred uniformity is not always satisfied, we have the weaker extensionality. Relevance is not satisfied, nor its weakening, core-retainment.

Proposition 3.6. [SMRW18] *The $\text{pmc}_{\mathbf{g}}^{\text{Cn}^*}$ operation satisfies extensionality, but does not satisfy core-retainment.*

Depending on some properties of Cn^* , we can assign additional properties to the operation. If Cn^* satisfies inclusion and subclassicality, particularly, then logical inclusion and uniformity will hold.

Observation 3.7. [SMRW18] *If Cn^* satisfies*

- *subclassicality, then an operation satisfying inclusion* also satisfies logical inclusion;*
- *inclusion, then an operation satisfying uniformity* also satisfies uniformity;*
- *subclassicality and inclusion, then an operation satisfying relevance* also satisfies logical relevance.*

*Proof.*⁹

- If c^{Cn^*} satisfies inclusion*, assuming that Cn^* is subclassical, we have:

$$\begin{aligned} c^{\text{Cn}^*}(B, \alpha) &\subseteq \text{Cn}^*(B) && \text{(inclusion* of } \text{Cn}^*) \\ c^{\text{Cn}^*}(B, \alpha) &\subseteq \text{Cn}(B) && \text{(subclassicality of } \text{Cn}^*) \\ \text{Cn}(c^{\text{Cn}^*}(B, \alpha)) &\subseteq \text{Cn}(\text{Cn}(B)) && \text{(monotonicity of } \text{Cn}) \\ \text{Cn}(c^{\text{Cn}^*}(B, \alpha)) &\subseteq \text{Cn}(B) && \text{(idempotence of } \text{Cn}) \end{aligned}$$

and thus c^{Cn^*} satisfies logical inclusion.

- If Cn^* satisfies inclusion and c^{Cn^*} satisfies uniformity*, then (from inclusion of Cn^*) every $B' \subseteq B$ is also a subset of $\text{Cn}^*(B)$, and for each such B' , uniformity* and uniformity state exactly the same. Therefore, c^{Cn^*} satisfies uniformity.
- Assuming that Cn^* satisfies inclusion and subclassicality, if c^{Cn^*} satisfies relevance*, then for every $\beta \in B \setminus c^{\text{Cn}^*}(B, \alpha)$, we have $\beta \in \text{Cn}^*(B) \setminus c^{\text{Cn}^*}(B, \alpha)$ (from inclusion of Cn^*). Because of relevance*, there must be some B' such that $c^{\text{Cn}^*}(B, \alpha) \subseteq B' \subseteq \text{Cn}^*(B)$ and $\alpha \in \text{Cn}(B' \cup \{\beta\}) \setminus \text{Cn}(B')$. Since Cn^* is subclassical, such B' is also a subset of $\text{Cn}(B)$. Hence, c^{Cn^*} satisfies logical relevance. ■

From this observation it follows that Cn^* being subclassical is a sufficient condition for $\text{pmc}_{\mathbf{g}}^{\text{Cn}^*}$ to be properly called a pseudo-contraction (success is granted by the partial meet).

The fact that kernel contraction does not satisfy relevance causes it to violate relative closure, an important property when it comes to rational contractions. The Cn^* partial meet pseudo-contraction also violates relevance, but it still preserves relative closure.

Proposition 3.8. [San16] *If Cn^* satisfies inclusion, the $\text{pmc}_{\mathbf{g}}^{\text{Cn}^*}$ operation satisfies relative closure.*

Cn^* partial meet pseudo-contraction operates in an intermediate level between the complete syntactic independence of classical Belief Revision and complete syntactic dependence of belief base operation. Uniformity* describes the kind of syntactic independence that can be expected for the proposed operation. This characteristic has important consequences. For example, vacuity

⁹This proof was omitted in [SMRW18], but we provide it here because this result will also be used in the next section.

(a principle of minimal change which states that, if you do not believe in the input sentence, its contraction will not change your beliefs) is not satisfied. This violation is related to the satisfaction of another property, which we call enforced closure*, that illustrates this balance on syntactic independence.

$$\text{(enforced closure}^*) \ c(B, \alpha) = \text{Cn}^*(c(B, \alpha)).$$

Proposition 3.9. [San16] *If Cn^* is Tarskian and satisfies subclassicality, an operation that satisfies inclusion* and relevance* also satisfies enforced closure*.*

Enforced closure* forces the base to be closed, even when the input is not in the original set of beliefs. Nevertheless, the $\text{pmc}_g^{\text{Cn}^*}$ operation at least satisfies a weaker variant of vacuity:

$$\text{(vacuity}^*) \ \text{If } \alpha \notin \text{Cn}(B), \text{ then } c(B, \alpha) = \text{Cn}^*(B).$$

Proposition 3.10. [San16] *If Cn^* satisfies subclassicality, an operation that satisfies inclusion* and relevance*¹⁰ also satisfies vacuity*.*

In the next section, we will present the analogous pseudo-contraction construction based on kernel.

3.2 Cn^* kernel pseudo-contraction

We have shown a pseudo-contraction operation that is defined as a partial meet contraction starting from the expanded set $\text{Cn}^*(B)$. In this section, we will present a similar construction that also expands the belief base before contracting, but uses kernel rather than partial meet. The contents of this section have not been published, unless where noted otherwise.

The construction of this new operation can be thought of as a kernel contraction of a previously expanded set. We will start with its definition:

Definition 3.11 (Cn^* kernel pseudo-contraction). Let B be a set of sentences, Cn^* a consequence relation and \mathbf{f} an incision function for $\text{Cn}^*(B)$. The Cn^* kernel pseudo-contraction of B by a sentence α , denoted by $\text{kc}_{\mathbf{f}}^{\text{Cn}^*}(B, \alpha)$, is such that, for all sentences α :

$$\text{kc}_{\mathbf{f}}^{\text{Cn}^*}(B, \alpha) = \text{Cn}^*(B) \setminus \mathbf{f}(\text{Ker}[\text{Cn}^*(B), \alpha]).$$

The following examples illustrate this construction:

Example 3.12. In the problem discussed in Example 3.1, let us consider again the consequence operator $\text{Cn}^*_{\text{break}}$ defined in Example 3.4. We have

$$\text{Ker}[\text{Cn}^*_{\text{break}}(B), s \text{ hasMainColour white}] = \{\{\beta_1, \beta_2, \beta_3\}, \{\beta_1, \beta_2, \beta'_3, \beta''_3\}\},$$

where

$$\begin{aligned} \beta_1 &= \text{Swedish} \sqsubseteq \text{European}, \\ \beta_2 &= \text{European} \sqcap \text{Swan} \sqsubseteq \exists \text{hasMainColour}.\{\text{white}\}, \\ \beta_3 &= s : \text{Swan} \sqcap \text{Swedish}, \\ \beta'_3 &= s : \text{Swan} \text{ and} \\ \beta''_3 &= s : \text{Swedish}. \end{aligned}$$

If the incision function \mathbf{f} is such that $\mathbf{f}(\text{Ker}[\text{Cn}^*_{\text{break}}(B), s \text{ hasMainColour white}]) = \{\beta_3, \beta''_3\}$, then $\text{kc}_{\mathbf{f}}^{\text{Cn}^*_{\text{break}}}(B, s \text{ hasMainColour white}) = (B \setminus \beta_3) \cup \{\beta'_3\}$, i.e., the pseudo-contraction replaces β_3 with its weaker version β'_3 .

¹⁰As we will show in the next section (Proposition 3.20), this proposition still holds if relevance* is replaced with a specific weaker condition.

Example 3.13. [RW08b, SMRW18, adapted] Consider the following knowledge base, which states that Cleopatra has a son and a daughter.

- Concepts: *Person*, *Man* and *Woman*.
- Role: *hasChild* (domain: *Person*; range: *Person*).
- Individuals (assumed different): *cleopatra*, c_1 and c_2 .
- TBox axioms: $Man \sqsubseteq Person$ and $Woman \sqsubseteq Person$.
- ABox axioms: $c_1 : Man$, $c_2 : Woman$, $cleopatra : Woman$, $cleopatra \text{ hasChild } c_1$ and $cleopatra \text{ hasChild } c_2$.

Let B be the knowledge base above, and let α be the sentence $cleopatra : \exists \text{hasChild}.Man$. Let Cn^*_{sup} be a consequence operator that adds a sentence $a : D$ for every individual a and every concept D such that $a : C$ and $C \sqsubseteq D$ are in the input set. We have

$$Cn^*_{\text{sup}}(B) = B \cup \{c_1 : Person, c_2 : Person, cleopatra : Person\}$$

and

$$\text{Ker}[Cn^*_{\text{sup}}(B), \alpha] = \{c_1 : Man, cleopatra \text{ hasChild } c_1\}.$$

If we have an incision function f such that $f(\text{Ker}[Cn^*_{\text{sup}}(B), \alpha]) = \{c_1 : Man\}$, then the operation removes our belief that c_1 is a man whilst preserving the (previously implicit) belief that c_1 is a person.

In order to characterise this operation, we will need a starred version of core-retainment:

(core-retainment*) If $\beta \in Cn^*(B) \setminus c(B, \alpha)$, then there is some $B' \subseteq Cn^*(B)$ such that $\alpha \in Cn(B' \cup \{\beta\}) \setminus Cn(B')$.

The representation theorem follows.

Theorem 3.14 (Cn* kernel pseudo-contraction: representation theorem). *If Cn* satisfies monotonicity, then an operation is a Cn* kernel pseudo-contraction if and only if it satisfies success, inclusion*, core-retainment* and uniformity*.*

Proof.

Construction-to-postulates:

- Success: We prove by contradiction. Assume that $\alpha \in Cn(\text{kc}_f^{Cn^*}(B, \alpha))$. Since Cn is compact, there is some non-empty $X \in \text{Ker}[Cn^*(B) \setminus f(\text{Ker}[Cn^*(B), \alpha]), \alpha]$. Such X is also an inclusion-minimal subset of $Cn^*(B)$ implying α , i.e., $X \in \text{Ker}[Cn^*(B), \alpha]$. Moreover, as $X \subseteq Cn^*(B) \setminus f(\text{Ker}[Cn^*(B), \alpha])$, we have $f(\text{Ker}[Cn^*(B), \alpha]) \cap X = \emptyset$, which violates the definition of incision function.
- Inclusion*: Follows directly from the definition.
- Core-retainment*: If $\beta \in Cn^*(B) \setminus \text{kc}_f^{Cn^*}(B, \alpha)$, then this means that $\beta \in Cn^*(B) \setminus (Cn^*(B) \setminus f(\text{Ker}[Cn^*(B), \alpha]))$, then $\beta \in f(\text{Ker}[Cn^*(B), \alpha])$. Hence, β is an element of some $X \in \text{Ker}[Cn^*(B), \alpha]$. The set $B' := X \setminus \{\beta\}$ is such that $B' \subseteq Cn^*(B)$, and since X is an α -kernel of $Cn^*(B)$, we have that $\alpha \in Cn(B' \cup \{\beta\}) \setminus Cn(B')$.
- Uniformity*: Suppose that, for all $B' \subseteq Cn^*(B)$, $\alpha \in Cn(B')$ if and only if $\beta \in Cn(B')$. We want to prove that $\text{kc}_f^{Cn^*}(B, \alpha) = \text{kc}_f^{Cn^*}(B, \beta)$. It is enough to show that $\text{Ker}[Cn^*(B), \alpha] = \text{Ker}[Cn^*(B), \beta]$, from which the result follows due to the definition of $\text{kc}_f^{Cn^*}$. We prove by contradiction. Assume that there is some $X \in \text{Ker}[Cn^*(B), \alpha] \setminus \text{Ker}[Cn^*(B), \beta]$ (for the

opposite case, the proof works by exchanging α and β). Since $X \subseteq \text{Cn}^*(B)$ and X is not a β -kernel of $\text{Cn}^*(B)$, then either $\beta \notin \text{Cn}(X)$ or there is some $X' \subsetneq X$ such that $\beta \in \text{Cn}(X')$. The first case implies that $\alpha \notin \text{Cn}(X)$, which is not possible because X is an α -kernel of $\text{Cn}^*(B)$. The second case is also impossible: since $X' \subseteq \text{Cn}^*(B)$ and $\beta \in \text{Cn}(X')$, we have that $\alpha \in \text{Cn}(X')$, which cannot hold because X is an inclusion-minimal α -implying subset of $\text{Cn}^*(B)$.

Postulates-to-construction: This part of the proof is similar to the proof of the corresponding theorem in [Han94]. We need to show that an operation c^{Cn^*} that satisfies success, inclusion*, core-retainment* and uniformity* is a Cn^* kernel pseudo-contraction. Let us define \mathbf{f} as

$$\mathbf{f}(\text{Ker}[\text{Cn}^*(B), \alpha]) := \text{Cn}^*(B) \setminus c^{\text{Cn}^*}(B, \alpha).$$

We shall prove that (i) \mathbf{f} is well defined, (ii) \mathbf{f} is an incision function for $\text{Cn}^*(B)$ and (iii) the operation $\text{kc}_{\mathbf{f}}^{\text{Cn}^*}$ is equivalent to c^{Cn^*} :

- (i) We have to show that the result of \mathbf{f} is always the same for a fixed input. Suppose that α_1 and α_2 are such that $\text{Ker}[\text{Cn}^*(B), \alpha_1] = \text{Ker}[\text{Cn}^*(B), \alpha_2]$. Consider any $B' \in \text{Cn}^*(B)$. If $\alpha_1 \in \text{Cn}(B')$, from compactness of Cn we can consider a finite inclusion-minimal α_1 -implying set $B'' \subseteq B'$, i.e. $B'' \in \text{Ker}[\text{Cn}^*(B), \alpha_1]$, thus $B'' \in \text{Ker}[\text{Cn}^*(B), \alpha_2]$ and $\alpha_2 \in \text{Cn}(B'')$, which implies (due to monotonicity of Cn) that $\alpha_2 \in \text{Cn}(B')$. Similarly, $\alpha_2 \in \text{Cn}(B')$ implies $\alpha_1 \in \text{Cn}(B')$. Hence, we have that $\alpha_1 \in \text{Cn}(B')$ if and only if $\alpha_2 \in \text{Cn}(B')$. By uniformity*, $c^{\text{Cn}^*}(B, \alpha_1) = c^{\text{Cn}^*}(B, \alpha_2)$, and from the definition of \mathbf{f} we conclude that $\mathbf{f}(\text{Ker}[\text{Cn}^*(B), \alpha_1]) = \mathbf{f}(\text{Ker}[\text{Cn}^*(B), \alpha_2])$.
- (ii) To prove that \mathbf{f} is an incision function, we must show that (1) $\mathbf{f}(\text{Ker}[\text{Cn}^*(B), \alpha]) \subseteq \bigcup \text{Ker}[\text{Cn}^*(B), \alpha]$ and (2) $\mathbf{f}(\text{Ker}[B, \alpha]) \cap X \neq \emptyset$ for every $X \in \text{Ker}[B, \alpha] \setminus \{\emptyset\}$.
 - (1) Consider any $\beta \in \mathbf{f}(\text{Ker}[\text{Cn}^*(B), \alpha])$. We will show that $\beta \in \bigcup \text{Ker}[\text{Cn}^*(B), \alpha]$. From the definition of \mathbf{f} , we have that $\beta \in \text{Cn}^*(B) \setminus c^{\text{Cn}^*}(B, \alpha)$. Since c^{Cn^*} satisfies core-retainment*, there must be some $B' \subseteq \text{Cn}^*(B)$ such that $\alpha \in \text{Cn}(B' \cup \{\beta\}) \setminus \text{Cn}(B')$. By compactness of Cn , there is some (finite) inclusion-minimal α -implying subset $B'' \subseteq B' \cup \{\beta\}$ such that $\alpha \in \text{Cn}(B'')$, and β must be in B'' because Cn is monotonic and $\alpha \notin \text{Cn}(B')$. As $B' \subseteq \text{Cn}^*(B)$ and $\beta \in \text{Cn}^*(B)$, the set B'' is also subset of $\text{Cn}^*(B)$. Hence, $B'' \in \text{Ker}[\text{Cn}^*(B), \alpha]$, and since $\beta \in B''$, we conclude that $\beta \in \bigcup \text{Ker}[\text{Cn}^*(B), \alpha]$.
 - (2) Take any non-empty $X \in \text{Ker}[\text{Cn}^*(B), \alpha]$. By the definition of kernel, it must be the case that $\alpha \notin \text{Cn}(\emptyset)$ and $\alpha \in \text{Cn}(X)$. From success, we have that $\alpha \notin \text{Cn}(c^{\text{Cn}^*}(B, \alpha))$, which implies (due to inclusion of Cn) that $\alpha \notin c^{\text{Cn}^*}(B, \alpha)$. Then, $X \not\subseteq c^{\text{Cn}^*}(B, \alpha)$, and there must be some $\beta \in X \setminus c^{\text{Cn}^*}(B, \alpha)$. Since $X \subseteq \text{Cn}^*(B)$, we have that $\beta \in \text{Cn}^*(B) \setminus c^{\text{Cn}^*}(B, \alpha)$, that is, $\beta \in \mathbf{f}(\text{Ker}[\text{Cn}^*(B), \alpha])$. This concludes the proof that $\mathbf{f}(\text{Ker}[\text{Cn}^*(B), \alpha]) \cap X \neq \emptyset$.

(iii) We can rewrite $\text{kc}_{\mathbf{f}}^{\text{Cn}^*}(B, \alpha)$ as:

$$\begin{aligned} \text{kc}_{\mathbf{f}}^{\text{Cn}^*}(B, \alpha) &= \text{Cn}^*(B) \setminus \mathbf{f}(\text{Ker}[\text{Cn}^*(B), \alpha]) \\ &= \text{Cn}^*(B) \setminus (\text{Cn}^*(B) \setminus c^{\text{Cn}^*}(B, \alpha)) \\ &= \text{Cn}^*(B) \cap c^{\text{Cn}^*}(B, \alpha) && \text{(by inclusion* of } c^{\text{Cn}^*}\text{)} \\ &= c^{\text{Cn}^*}(B, \alpha). \end{aligned}$$

■

It comes as no surprise that every Cn^* partial meet pseudo-contraction is also a Cn^* kernel pseudo-contraction: core-retainment* is a generalised version of relevance*, and the other postulates are the same. We will now show how to obtain the explicit construction of a Cn^* kernel pseudo-contraction from a Cn^* partial meet pseudo-contraction. This will use the definition of an incision

function derived from a selection function, which was proposed by Falappa, Fermé and Kern-Isberner [FFK06]:

Definition 3.15 (Associated incision function [FFK06]). Let \mathbf{g} be a selection function for a set X . The function $\mathbf{f}_{\mathbf{g}}$ defined as

$$\mathbf{f}_{\mathbf{g}}(\text{Ker}[X, \alpha]) = X \setminus \bigcap \mathbf{g}(\text{Rem}[X, \alpha])$$

is the \mathbf{g} -associated incision function for X .

The function defined above is, indeed, an incision function:

Theorem 3.16. [FFK06] *The function $\mathbf{f}_{\mathbf{g}}$ (as in Definition 3.15) is an incision function for X .*

As mentioned earlier, Cn^* kernel pseudo-contraction subsumes Cn^* partial meet pseudo-contraction. The following proposition shows the explicit construction:

Proposition 3.17. *If $\text{pmc}_{\mathbf{g}}^{\text{Cn}^*}$ is a Cn^* partial meet pseudo-contraction, then it is equivalent to the Cn^* kernel pseudo-contraction $\text{kc}_{\mathbf{f}_{\mathbf{g}}}^{\text{Cn}^*}$.*

Proof. Let $\text{pmc}_{\mathbf{g}}^{\text{Cn}^*}$ be a Cn^* partial meet pseudo-contraction. We can rewrite it as follows:

$$\begin{aligned} \text{pmc}_{\mathbf{g}}^{\text{Cn}^*}(B, \alpha) &= \bigcap \mathbf{g}(\text{Rem}[\text{Cn}^*(B), \alpha]) \\ &= \text{Cn}^*(B) \setminus \left[\text{Cn}^*(B) \setminus \bigcap \mathbf{g}(\text{Rem}[\text{Cn}^*(B), \alpha]) \right] \\ &= \text{Cn}^*(B) \setminus \mathbf{f}_{\mathbf{g}}(\text{Ker}[X, \alpha]) \\ &= \text{kc}_{\mathbf{f}_{\mathbf{g}}}^{\text{Cn}^*}(B, \alpha), \end{aligned}$$

where $\mathbf{f}_{\mathbf{g}}$ is the incision function defined as in Definition 3.15 (for $X = \text{Cn}^*(B)$). ■

In general, not every kernel contraction is equivalent to a partial meet contraction: by taking Cn^* as the identity function, Cn^* partial meet and kernel pseudo-contractions become partial meet and kernel contractions for belief bases, which are not equivalent. Therefore, Cn^* kernel pseudo-contractions may not have the same properties as Cn^* partial meet pseudo-contractions shown in Section 3.1.

From Observation 3.1, we can see that a Cn^* kernel pseudo-contraction is indeed a pseudo-contraction as long as Cn^* is subclassical: success is one of its postulates, and the aforementioned observation shows that inclusion* (which is also a postulate) implies logical inclusion. If, besides being subclassical, Cn^* also satisfies inclusion, then the following property is satisfied by any Cn^* kernel pseudo-contraction:

(logical core-retainment) If $\beta \in B \setminus \text{c}(B, \alpha)$, then there is a B' such that $B' \subseteq \text{Cn}(B)$ and $\alpha \in \text{Cn}(B' \cup \{\beta\}) \setminus \text{Cn}(B')$.

Observation 3.18. *If Cn^* satisfies subclassicality and inclusion, then any operation that satisfies core-retainment* also satisfies logical core-retainment.*

Proof. Assuming that Cn^* satisfies inclusion and subclassicality, if c^{Cn^*} satisfies core-retainment*, then for every $\beta \in B \setminus \text{c}^{\text{Cn}^*}(B, \alpha)$, we have $\beta \in \text{Cn}^*(B) \setminus \text{c}^{\text{Cn}^*}(B, \alpha)$ (from inclusion of Cn^*). Because of core-retainment*, there must be some $B' \subseteq \text{Cn}^*(B)$ such that $\alpha \in \text{Cn}(B' \cup \{\beta\}) \setminus \text{Cn}(B')$. Since Cn^* is subclassical, such B' is also a subset of $\text{Cn}(B)$. Hence, c^{Cn^*} satisfies logical core-retainment. ■

As mentioned in Section 2.2, a kernel contraction does not necessarily satisfy relative closure, which is enough to conclude that this property does not always hold for Cn^* kernel pseudo-contractions. Nonetheless, smooth kernel contractions satisfy relative closure, and the same is true for smooth Cn^* kernel pseudo-contractions:

Proposition 3.19. *If the Cn^* kernel pseudo-contraction $c_{\mathbf{f}}^{\text{Cn}^*}$ is such that \mathbf{f} is smooth, then $c_{\mathbf{f}}^{\text{Cn}^*}$ satisfies relative closure.*

Proof. Let $c_{\mathbf{f}}^{\text{Cn}^*}$ be a Cn^* kernel pseudo-contraction, where \mathbf{f} is smooth. We will prove, by contradiction, that $c_{\mathbf{f}}^{\text{Cn}^*}$ satisfies relative closure. Assuming that this does not hold, there must be some sentence $\beta \in \left(B \cap \text{Cn}(c_{\mathbf{f}}^{\text{Cn}^*}(B, \alpha)) \right) \setminus c_{\mathbf{f}}^{\text{Cn}^*}(B, \alpha)$. Then, $\beta \in B \setminus c_{\mathbf{f}}^{\text{Cn}^*}(B, \alpha)$, i.e., $\beta \in B$ but $\beta \notin \text{Cn}^*(B) \setminus \mathbf{f}(\text{Ker}[\text{Cn}^*(B), \alpha])$. Since Cn^* satisfies inclusion, β must be in $\text{Cn}^*(B)$, so β must be in $\mathbf{f}(\text{Ker}[\text{Cn}^*(B), \alpha])$. Let $B' := B \setminus \mathbf{f}(\text{Ker}[\text{Cn}^*(B), \alpha])$. The set B' is such that $B' \subseteq B$, and from inclusion of Cn^* we have that $B' \subseteq \text{Cn}^*(B)$. Also, as β is in both $\text{Cn}^*(B)$ and $\mathbf{f}(\text{Ker}[\text{Cn}^*(B), \alpha])$, the set $\text{Cn}^*(B) \cap \mathbf{f}(\text{Ker}[\text{Cn}^*(B), \alpha])$ is non-empty. From the smoothness of \mathbf{f} , the set $B' \cap \mathbf{f}(\text{Ker}[\text{Cn}^*(B), \alpha])$ must not be empty. This is a contradiction, because the definition of B' implies that it cannot contain any element of $\mathbf{f}(\text{Ker}[\text{Cn}^*(B), \alpha])$. ■

Cn^* kernel pseudo-contraction does not satisfy vacuity: as shown in Section 3.1, it is not satisfied by Cn^* partial meet pseudo-contraction. This means that the result of a Cn^* kernel pseudo-contraction may be different from the original set even when the input sentence was not entailed by the initial belief base. Nonetheless, a weaker version of the property — vacuity* — is satisfied:

Proposition 3.20. *If Cn^* satisfies subclassicality, an operation that satisfies inclusion* and core-retainment* also satisfies vacuity*.*

Proof. Let c be an operation that satisfies inclusion* and core-retainment*, and assume that Cn^* is subclassical. Let B be a set of sentences and α be any sentence such that $\alpha \notin \text{Cn}(B)$. We want to prove that $c(B, \alpha) = \text{Cn}^*(B)$. Inclusion* gives us $c(B, \alpha) \subseteq \text{Cn}^*(B)$, so we only have to show that $c(B, \alpha) \supseteq \text{Cn}^*(B)$, i.e. the set $\text{Cn}^*(B) \setminus c(B, \alpha)$ is empty. We will prove by contradiction. Assume that there is some $\beta \in \text{Cn}^*(B) \setminus c(B, \alpha)$. As c satisfies core-retainment*, there must be some $B' \subseteq \text{Cn}^*(B)$ such that $\alpha \in \text{Cn}(B' \cup \{\beta\}) \setminus \text{Cn}(B')$. Since $B' \subseteq \text{Cn}^*(B)$ and $\beta \in \text{Cn}^*(B)$, we have $B' \cup \{\beta\} \subseteq \text{Cn}^*(B)$, and from subclassicality of Cn^* , this implies that $B' \cup \{\beta\} \subseteq \text{Cn}(B)$. Because Cn satisfies monotonicity, we have $\text{Cn}(B' \cup \{\beta\}) \subseteq \text{Cn}(\text{Cn}(B))$, and using idempotence of Cn , we obtain $\text{Cn}(B' \cup \{\beta\}) \subseteq \text{Cn}(B)$. This is a contradiction, because $\alpha \in \text{Cn}(B' \cup \{\beta\})$ but $\alpha \notin \text{Cn}(B)$. ■

3.3 Pseudo-contraction plug-in for Protégé: a prototype

We have implemented a prototype of a Protégé plug-in that computes Cn^* (partial meet and kernel) pseudo-contractions. The algorithms for obtaining remainder and kernel sets were adapted from Guimarães' repository [Gui20]¹¹.

The plug-in provides a tab which can be added to Protégé interface from the menu (*Window* → *Tabs* → *Pseudo-contraction*), as shown in Figure 3.2.

After the tab is added, the user can type the sentence to be contracted using Manchester syntax [HDG⁺06] and the traditional notation of Description Logics is displayed below the input field, as shown in Figure 3.3.

Besides the sentence α to be contracted, the user can choose:

- Cn^* : the types of consequences that will be generated by the reasoner before the operation;
- construction: partial meet or kernel;
- strategies: methods used to compute remainder and kernel sets (see [Rei87, Was00, KPHS07, CW15, Gui20]).

¹¹<https://gitlab.com/rfguimaraes/owl-change>

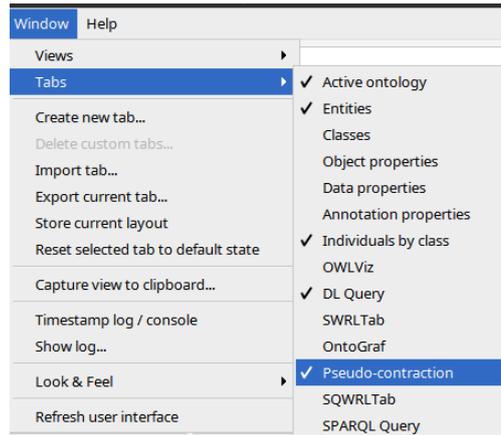


Figure 3.2: Screenshot of Protégé window showing the menu that enables the insertion of the pseudo-contraction tab into the interface.

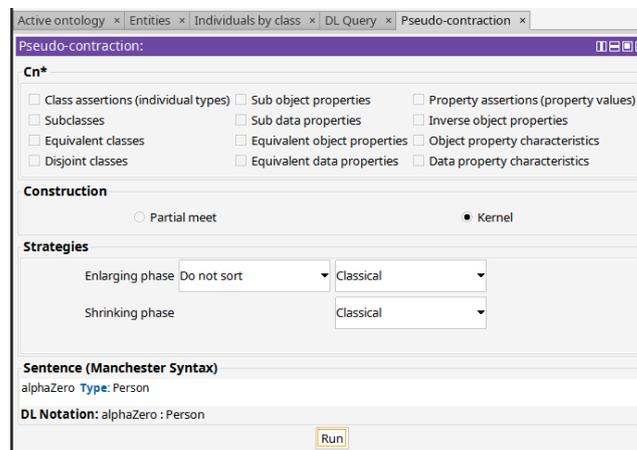


Figure 3.3: Screenshot of Protégé window with the pseudo-contraction tab.

After the user clicks on *Run*, the program obtains $Cn^*(\mathcal{O})$ from the original ontology \mathcal{O} and computes $Rem[Cn^*(\mathcal{O}), \alpha]$ or $Ker[Cn^*(\mathcal{O}), \alpha]$, according to the chosen construction. The remainders or kernels are then shown on a dialogue window, where they can be chosen by the user. Figure 3.4 shows the kernel set of Example 2.17 and the incision function given in the example. The remainder sets of the problem discussed in Example 3.1 and Example 3.4 with the corresponding selection functions are shown in Figure 3.5 and Figure 3.6 (in the latter case, there are more remainders than in the example because the consequence generator *Class assertions* is more general than Cn^*_{break} and generates the additional sentence $s : European$). The kernel set and the incision function discussed in Example 3.12 are depicted in Figure 3.7 (it contains more kernels for the aforesaid reason).

In all cases, the sentences that will be removed are displayed in red. To avoid cluttering the window, *Declaration* sentences and obvious tautologies (such as $a : \top$ and $C \sqsubseteq \top$) are omitted. After a click on the button *Execute operation*, the plug-in transforms the active ontology \mathcal{O} into $pmc_g^{Cn^*}(\mathcal{O}, \alpha)$ or $kc_f^{Cn^*}(\mathcal{O}, \alpha)$, where Cn^* is determined by the selected sentence generators and the chosen remainders or kernel elements define the function g or f .

The plug-in is written in Java 8 and supports Protégé 5.5.0, which is the latest version at the time of writing. It uses OWL API¹² 4.2.5 to manipulate OWL objects¹³. The source code is available on GitLab¹⁴.

¹²<http://owlcs.github.io/owlapi/>

¹³Version 5.1.17 of OWL API is already available, but it is not supported by Protégé yet, which is why we had to use a previous version.

¹⁴<https://gitlab.com/viniciusbm/pseudo-contraction-protege-plugin>

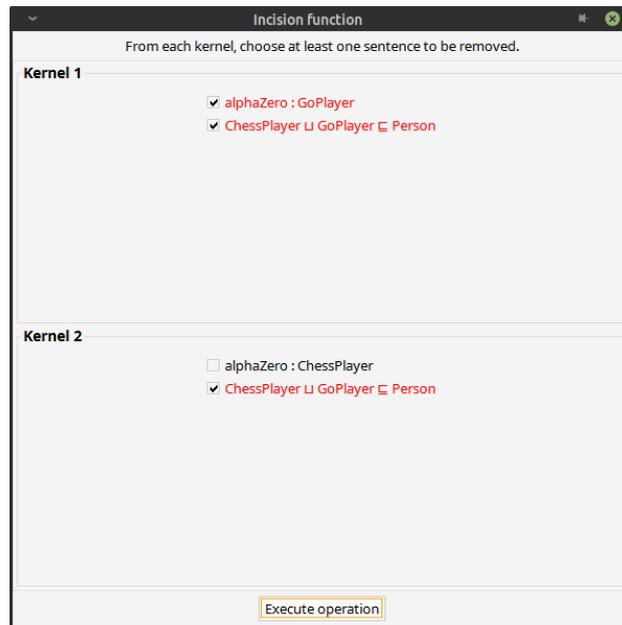


Figure 3.4: Screenshot of Protégé window with the kernel set of Example 2.17.

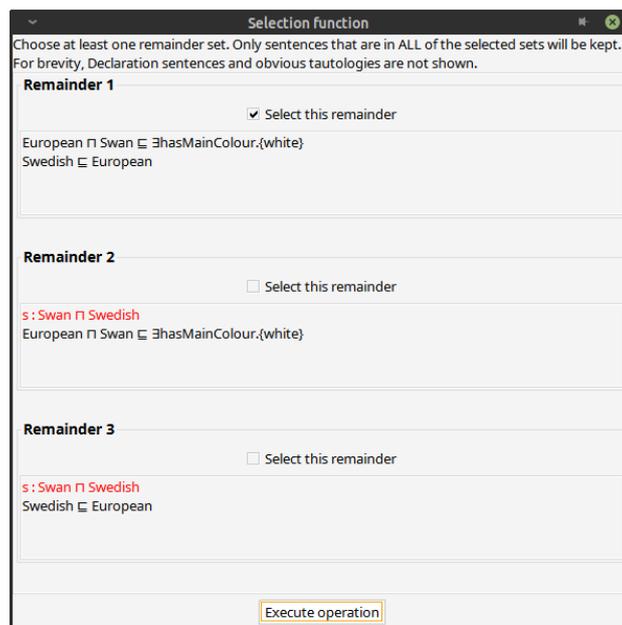


Figure 3.5: Screenshot of Protégé window with the remainder set of Example 3.1.

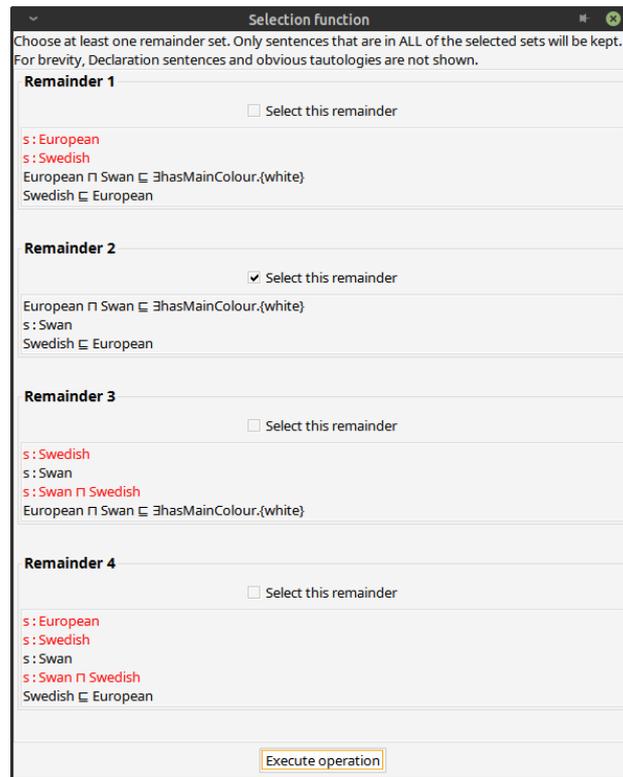


Figure 3.6: Screenshot of Protégé window with the remainder set of Example 3.4.

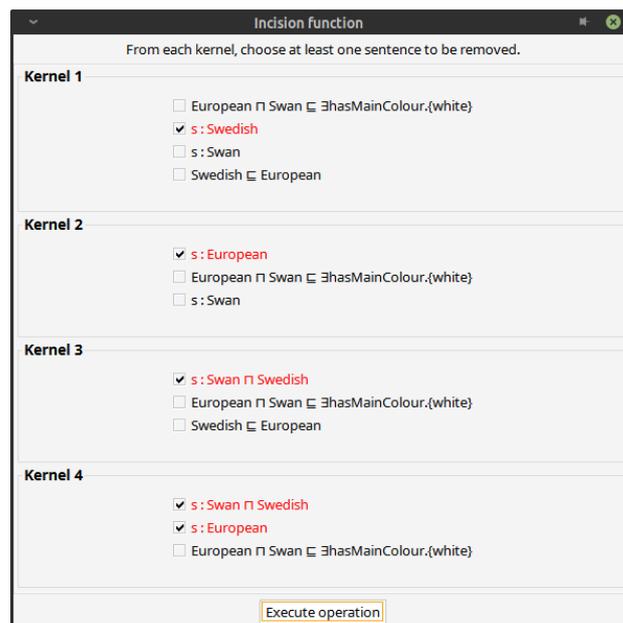


Figure 3.7: Screenshot of Protégé window with the kernel set of Example 3.12.

3.4 Other operations

We will finish the chapter with a brief introduction to two-place (partial meet and kernel) pseudo-contractions and a proposal of a pseudo-revision definition. Only the excerpt about two-place partial meet pseudo-contraction has been published.

3.4.1 Two-place pseudo-contractions

In this subsection, we present another kind of Cn^* [partial meet and kernel] pseudo-contraction, using consequence operators (denoted by Cn^{**}) that also take into account the input formula. We will start with an example that shows the rationale for such complex consequence operators.

Example 3.21. Consider, for the last time, the example we presented in [Example 3.1](#) (continued in [Example 3.4](#) and [Example 3.12](#)). The consequence operator $\text{Cn}^*_{\text{break}}$ did exactly what we would intuitively expect in that example because we had only one sentence of the form $a : C_1 \sqcap \dots \sqcap C_n$, which would be removed by classical contractions. However, such $\text{Cn}^*_{\text{break}}$ may add too many unnecessary sentences to the set. Let $B' = \{a_1 : C_1 \sqcap D_1 \sqcap E_1, a_2 : C_2 \sqcap D_2 \sqcap E_2, a_3 : C_3 \sqcap D_3 \sqcap E_3, a_4 : C_4 \sqcap D_4 \sqcap E_4\}$, from which we want to contract the sentence $a : C_1$. A classical contraction would remove the sentence $a : C_1 \sqcap D_1 \sqcap E_1$, and we would lose the assertions that a_1 is an instance of D_1 and E_1 . A pseudo-contraction allows the insertion of sentences such as $a_1 : D_1$ and $a_1 : E_1$. If we use $\text{Cn}^*_{\text{break}}$, the expanded set would also contain the sentences $a_j : C_j$, $a_j : D_j$ and $a_j : E_j$ (for $j \in \{2, 3, 4\}$), which are redundant and can make the set harder to work with computationally. Ideally, we should expand our base using a consequence operator that would only add relevant sentences. In order to decide whether adding a sentence would be redundant, the consequence operator must consider the sentence that we want to contract, which is precisely what two-place consequence operators do.

The definition of a partial meet pseudo-contraction that uses a two-place consequence operator follows.

Definition 3.22 (Two-place Cn^{**} partial meet pseudo-contraction [[SMRW18](#)]).¹⁵ Let B be a set of sentences, $\text{Cn}^{**} : 2^{\mathcal{L}} \times \mathcal{L} \rightarrow 2^{\mathcal{L}}$ and \mathbf{g} a selection function for $\text{Cn}^{**}(B, \alpha)$. The *two-place Cn^{**} partial meet pseudo-contraction* of B by a sentence α , denoted by $\text{tpPMC}_{\mathbf{g}}^{\text{Cn}^{**}}(B, \alpha)$, is such that, for all sentences α :

$$\text{tpPMC}_{\mathbf{g}}^{\text{Cn}^{**}}(B, \alpha) = \bigcap \mathbf{g}(\text{Rem}[\text{Cn}^{**}(B, \alpha), \alpha]).$$

This operation generalises Cn^* partial meet pseudo-contractions ([Definition 3.3](#)): if, for any set B , the result of $\text{Cn}^{**}(B, \alpha)$ is the same for every sentence α , then we define $\text{Cn}^*(B) := \text{Cn}^{**}(B, \alpha)$ (which does not depend on α), and we have a Cn^* partial meet pseudo-contraction:

Proposition 3.23. [[SMRW18](#)] *Every Cn^* partial meet pseudo-contraction is a two-place Cn^{**} partial meet pseudo-contraction.*

An example of the use of two-place Cn^{**} partial meet pseudo-contraction is the idea of *local change*. In [[HW02](#)], Hansson and Wassermann proposed a *local* consequence operator which looks at the relevant part of the belief base with respect to a certain sentence and extracts its consequences. It is defined as $\text{Cn}^{**}_L(B, \alpha) := \text{Cn}(\text{Comp}[B, \alpha])$, where $\text{Comp}[B, \alpha]$ is the set of *compartments around a formula*, which are the elements of the minimal consistent subsets of B that imply either α or its negation:

$$\text{Comp}[B, \alpha] := \begin{cases} \emptyset, & \text{if } \{\alpha, \neg\alpha\} \cap \text{Cn}(\emptyset) \neq \emptyset; \\ \bigcup \left((\text{Ker}[B, \alpha] \cup \text{Ker}[B, \neg\alpha]) \setminus \text{Ker}[B, \perp] \right), & \text{otherwise.} \end{cases}$$

As Cn^{**}_L is meant to generate only the inferences that are relevant to α , it is easy to see that this operator is not Tarskian, as it does not satisfy inclusion. Based on this idea, a similar two-place

¹⁵In [[SMRW18](#)], this operation was called ‘two-place Cn^* pseudo-contraction’. We added ‘partial meet’ because we also propose a kernel construction, and we wrote an extra star for two-place consequence operators.

consequence operator was proposed in [SMRW18], ensuring the inclusion of the base: $\text{Cn}^{**}_L(B, \alpha) := B \cup \text{Cn}(\text{Comp}[B, \alpha])$. This weak closure operator expands the original belief base with respect only to beliefs that can be considered relevant to the sentence being contracted. It satisfies inclusion, monotonicity and subclassicality, but does not satisfy idempotence [SMRW18], implying that Proposition 3.9 is not applicable to Cn^{**}_L .

Since two-place Cn^{**} partial meet pseudo-contractions are generalisations of Cn^* partial meet pseudo-contractions, it is natural that some properties of the latter do not hold for the former. Particularly, the two-place version satisfies success, inclusion* and relevance* (the proof for this fact is analogous to the one given in Theorem 3.5), but uniformity* is lost [SMRW18].

The construction presented in Definition 3.22 can be seen as a partial meet (base) contraction of the set $\text{Cn}^{**}(B, \alpha)$ by α , i.e., $\text{tppmc}_g^{\text{Cn}^{**}}(B, \alpha) = \text{pmc}_g(\text{Cn}^{**}(B, \alpha), \alpha)$. Similarly, we can define $\text{tpkc}_f^{\text{Cn}^{**}}(B, \alpha) = \text{kc}_f(\text{Cn}^{**}(B, \alpha), \alpha)$:

Definition 3.24 (Two-place Cn^{**} kernel pseudo-contraction). Let B be a set of sentences, $\text{Cn}^{**} : 2^{\mathcal{L}} \times \mathcal{L} \rightarrow 2^{\mathcal{L}}$ and f be an incision function for $\text{Cn}^{**}(B, \alpha)$. The *two-place Cn^{**} kernel pseudo-contraction* of B by a sentence α , denoted by $\text{tpkc}_f^{\text{Cn}^{**}}(B, \alpha)$, is such that, for all sentences α :

$$\text{tpkc}_f^{\text{Cn}^{**}}(B, \alpha) = \text{Cn}^{**}(B, \alpha) \setminus f(\text{Ker}[\text{Cn}^{**}(B, \alpha), \alpha]).$$

For the same reasons outlined above (see the paragraph above Proposition 3.23), this operation generalises Cn^* kernel pseudo-contractions:

Proposition 3.25. *Every Cn^* kernel pseudo-contraction is a two-place Cn^{**} kernel pseudo-contraction.*

Unsurprisingly, the more general version fails to satisfy some properties of the other operation: in particular, two-place Cn^{**} kernel pseudo-contractions satisfy success, inclusion* and core-retainment* (which can be proved in a similar way to Theorem 3.14), but uniformity* does not hold.

The construction shown in Proposition 3.17 can also be adapted to two-place pseudo-contractions, by replacing $\text{Cn}^*(B)$ with $\text{Cn}^{**}(B, \alpha)$; therefore, every two-place Cn^{**} partial meet pseudo-contraction is also a two-place Cn^{**} kernel pseudo-contraction.

Figure 3.8 shows the inclusion relationships¹⁶ among the Belief Revision contraction and pseudo-contraction operations we have discussed.

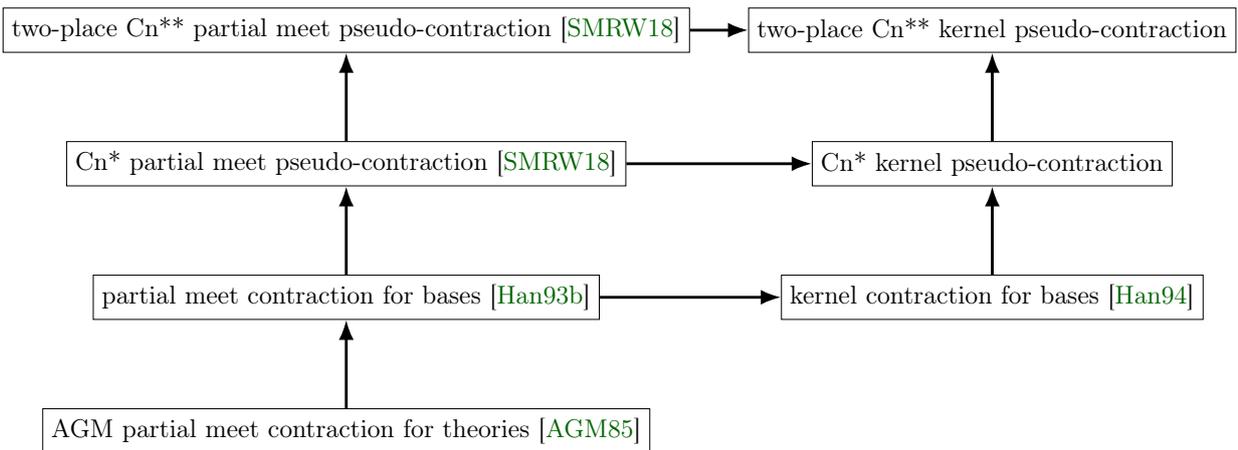


Figure 3.8: ‘Subtype’ relations among the Belief Revision (pseudo-)contraction operations discussed in this text. An arrow $u \rightarrow v$ means that all operations of type u are also operations of type v .

¹⁶As a matter of fact, the figure shows the transitive reduction of the actual graph (which is its transitive closure), because redundant arrows have been omitted.

3.4.2 Pseudo-revisions and adapted identities

The set of postulates required by a pseudo-contraction is almost a subset of the set of postulates for base contraction, where the only exception is the replacement of inclusion with a less constraining postulate called logical inclusion. Similarly, it is possible to define a new type of operation called *pseudo-revision*, which generalises revision and satisfies the following postulates:

(success) $\alpha \in r(B, \alpha)$.

(logical inclusion) $r(B, \alpha) \subseteq \text{Cn}(B) \cup \{\alpha\}$.

(consistency) $\perp \notin \text{Cn}(r(B, \alpha))$, unless $\perp \in \text{Cn}(\{\alpha\})$.

Definition 3.26 (Pseudo-revision). An operation r is a *pseudo-revision* if r satisfies success, logical inclusion and consistency.

If the logic is closed under negation, we can construct a pseudo-contraction from a pseudo-revision and a pseudo-revision from a pseudo-contraction, as long as Cn is Tarskian:

Definition 3.27. Let c be a pseudo-contraction. The operation $r_{(c)}$ defined by $r_{(c)}(B, \alpha) = c(B, \neg\alpha) \cup \{\alpha\}$ is the *c-based pseudo-revision*.

Definition 3.28. Let r be a pseudo-revision. The operation $c_{(r)}$ defined by $c_{(r)}(B, \alpha) = r(B, \neg\alpha) \cap \text{Cn}(B)$ is the *r-based pseudo-contraction*.

These are adapted versions of Levi and Harper identities. Indeed, the resulting operations are pseudo-revisions and pseudo-contractions:

Proposition 3.29. *If c is a pseudo-contraction, then $r_{(c)}$ (Definition 3.27) is a pseudo-revision.*

Proof. Let c be a pseudo-contraction. The operation $r_{(c)}$ satisfies:

- success: follows straightforwardly from the definition;
- logical inclusion: from inclusion of Cn and logical inclusion of c , we have $c(B, \neg\alpha) \subseteq \text{Cn}(c(B, \neg\alpha)) \subseteq \text{Cn}(B)$, which implies that $c(B, \neg\alpha) \cup \{\alpha\} \subseteq \text{Cn}(B) \cup \{\alpha\}$;
- consistency: if $\perp \notin \text{Cn}(\{\alpha\})$, then $\neg\alpha \notin \text{Cn}(\emptyset)$, and success of c guarantees that $\neg\alpha \notin \text{Cn}(c(B, \neg\alpha))$, which implies that $c(B, \neg\alpha)$ is consistent (otherwise its consequences would contain every sentence in the language, including $\neg\alpha$); therefore, $c(B, \neg\alpha) \cup \{\alpha\}$ is consistent. ■

Proposition 3.30. *If r is a pseudo-revision, then $c_{(r)}$ (Definition 3.28) is a pseudo-contraction.*

Proof. Let r be a pseudo-revision. The operation $c_{(r)}$ satisfies:

- success: since Cn is monotonic, $\text{Cn}(r(B, \neg\alpha) \cap \text{Cn}(B)) \subseteq \text{Cn}(r(B, \neg\alpha))$, and as the latter set has $\neg\alpha$ as one of its elements (because r satisfies success and Cn satisfies inclusion), consistency implies that the sentence α cannot be in it, i.e. $\alpha \notin \text{Cn}(r(B, \neg\alpha))$, and thus $\alpha \notin \text{Cn}(r(B, \neg\alpha) \cap \text{Cn}(B))$;
- logical inclusion: from the monotonicity of Cn , we have $\text{Cn}(r(B, \neg\alpha) \cap \text{Cn}(B)) \subseteq \text{Cn}(\text{Cn}(B))$, which is equal to $\text{Cn}(B)$ because Cn is idempotent. ■

Starting from a pseudo-revision r , we can obtain the r -based pseudo-contraction $c_{(r)}$, from which we can obtain the pseudo-revision $r_{(c_{(r)})}$. It turns out that this pseudo-revision is precisely equal to the original one:

Proposition 3.31. *If r is a pseudo-revision, then the operation $r_{(c_{(r)})}$ is equal to r for every input.*

Proof. Let r be a pseudo-revision. We have:

$$\begin{aligned}
r_{(c_{(r)})}(B, \alpha) &= c_{(r)}(B, \neg\alpha) \cup \{\alpha\} && \text{(definition of } r_{(c_{(r)})}\text{)} \\
&= (r(B, \neg\neg\alpha) \cap \text{Cn}(B)) \cup \{\alpha\} && \text{(definition of } c_{(r)}\text{)} \\
&= (r(B, \alpha) \cap \text{Cn}(B)) \cup \{\alpha\} && \text{(elimination of } \neg\neg\text{)} \\
&= (r(B, \alpha) \cup \{\alpha\}) \cap (\text{Cn}(B) \cup \{\alpha\}) && \text{(set theory)} \\
&= r(B, \alpha) \cap (\text{Cn}(B) \cup \{\alpha\}) && \text{(because } \alpha \in r(B, \alpha)\text{)} \\
&= r(B, \alpha). && \text{(intersection of a subset)}
\end{aligned}$$

■

This equivalence does not occur if we do the same starting with a pseudo-contraction, but the resulting operation produces the same output as the initial one except for at most one sentence:

Proposition 3.32. *If c is a pseudo-contraction, then the operation $c_{(r_{(c)})}$ yields, for every input, either the same result as c or its result augmented by $\{\neg\alpha\}$, and the latter case occurs if $\neg\alpha \in \text{Cn}(B) \setminus c(B, \alpha)$.*

Proof. Let c be a pseudo-contraction. We have:

$$\begin{aligned}
c_{(r_{(c)})}(B, \alpha) &= r_{(c)}(B, \neg\alpha) \cap \text{Cn}(B) && \text{(definition of } c_{(r_{(c)})}\text{)} \\
&= (c(B, \neg\neg\alpha) \cup \{\neg\alpha\}) \cap \text{Cn}(B) && \text{(definition of } r_{(c)}\text{)} \\
&= (c(B, \alpha) \cup \{\neg\alpha\}) \cap \text{Cn}(B) && \text{(elimination of } \neg\neg\text{)} \\
&= (c(B, \alpha) \cap \text{Cn}(B)) \cup (\{\neg\alpha\} \cap \text{Cn}(B)) && \text{(set theory)} \\
&= c(B, \alpha) \cup (\{\neg\alpha\} \cap \text{Cn}(B)) && \text{(intersection of a subset)} \\
&= \begin{cases} c(B, \alpha) \cup \{\neg\alpha\}, & \text{if } \neg\alpha \in \text{Cn}(B) \setminus c(B, \alpha); \\ c(B, \alpha), & \text{otherwise.} \end{cases}
\end{aligned}$$

■

Chapter 4

Pseudo-contractions and gentle repairs

A gentle repair is an operation proposed by Baader et al. [BKNP18] to avoid removing too much information from an ontology during a repair. In this chapter, we investigate the relationship between pseudo-contraction operations and gentle repairs.

Both aim to avoid the complete deletion of sentences when replacing them with weaker versions is enough to prevent the entailment of the unwanted formula. We show the correspondence between concepts on both sides and investigate under which conditions they are equivalent. We propose a unified notation for the two approaches, which might contribute to the integration of the two areas. Most of the contents of this chapter were published in [MGSW19].

4.1 Ontology Repair

Ontology Repair consists in transforming an ontology so that it does not imply a certain formula. In what follows, we define the main concepts based on the presentation given by Baader et al. [BKNP18]. Consider that $\mathcal{O} = \langle \mathcal{O}_s, \mathcal{O}_r \rangle$ is an ontology consisting of a static and a refutable part (\mathcal{O}_s and \mathcal{O}_r , respectively), which are assumed to be disjoint.¹ The static part contains the axioms which we want to preserve when we repair the ontology, while the refutable part contains those which we are willing to give up if needed. We assume that the separation into a static and refutable part is given as part of the input, be it a decision of an ontology engineer or obtained via some (semi-)automatic process.

Definition 4.1 (Repair). Let $\mathcal{O} = \langle \mathcal{O}_s, \mathcal{O}_r \rangle$ be an ontology and let α be a sentence entailed by \mathcal{O} but not by \mathcal{O}_s . An ontology \mathcal{O}' is a *repair* of \mathcal{O} with respect to α if $\text{Cn}(\mathcal{O}_s \cup \mathcal{O}') \subseteq \text{Cn}(\mathcal{O}) \setminus \{\alpha\}$.

Classically, a repair consists of a subset of the refutable part of the ontology:

Definition 4.2 (Classical repair). A repair \mathcal{O}' of the ontology \mathcal{O} with respect to the sentence α is a *classical repair* if it is contained in \mathcal{O}_r .

And usually, we try to preserve as much knowledge as possible, looking for an optimal repair:

Definition 4.3 (Optimal repair). A repair \mathcal{O}' of the ontology \mathcal{O} with respect to the sentence α is an *optimal repair* if no other repair \mathcal{O}'' (of \mathcal{O} w.r.t. α) is such that $\text{Cn}(\mathcal{O}_s \cup \mathcal{O}') \subset \text{Cn}(\mathcal{O}_s \cup \mathcal{O}'')$.

An *optimal classical repair* is a classical repair which is optimal in the sense that there is no classical repair which contains it.

In order to find classical repairs, a construction based on the ideas of justifications and hitting sets can be used. Justifications are minimal subsets of an ontology that imply the unwanted sentence:

Definition 4.4 (Justification [Kal06]). Let $\mathcal{O} = \langle \mathcal{O}_s, \mathcal{O}_r \rangle$ be an ontology and α a sentence entailed by \mathcal{O} but not by \mathcal{O}_s . A *justification* for α in \mathcal{O} is an inclusion-minimal subset J of \mathcal{O}_r such that $\alpha \in \text{Cn}(\mathcal{O}_s \cup J)$. We will denote the set of all justifications for α in \mathcal{O} as $\text{Just}(\mathcal{O}, \alpha)$.

¹The notation $\langle \mathcal{O}_s, \mathcal{O}_r \rangle$ is meant to represent the set $\mathcal{O}_s \cup \mathcal{O}_r$ in a way that makes it possible to tell if a sentence is in the static part or in the refutable part.

Schlobach [Sch05] has proposed an algorithm to debug incoherent ontologies inspired by Reiter's hitting set tree [Rei87]. Other authors, such as Kalyanpur et al. [Kal06, KPHS07] and Horridge [Hor11], extended and generalised this algorithm to find all justifications for any given entailment.

Definition 4.5 (Hitting set [Rei87]). Given a set \mathcal{J} of justifications for a sentence in an ontology, a *hitting set* of \mathcal{J} is a set H of sentences contained in $\bigcup \mathcal{J}$ such that $H \cap J \neq \emptyset$ for every $J \in \mathcal{J}$.

Regarding the actual repair, done by removing at least one formula from each justification, a simple description is presented in [Algorithm 1](#) from Baader et al. [BKNP18]. We assume the existence of a function `Justifications`(\mathcal{O}, α) that computes `Just`(\mathcal{O}, α) and a function `MinimalHittingSet`(\mathcal{J}) that computes an inclusion-minimal hitting set of \mathcal{J} .

Algorithm 1: Classical repair algorithm

<p>Input: An ontology $\mathcal{O} = \langle \mathcal{O}_s, \mathcal{O}_r \rangle$ and a formula α Output: A classical repair \mathcal{O}' of \mathcal{O} w.r.t. α</p> <pre> 1 Function <i>ClassicalRepair</i>(\mathcal{O}, α) 2 $\mathcal{J} \leftarrow \text{Justifications}(\mathcal{O}, \alpha)$ 3 $H \leftarrow \text{MinimalHittingSet}(\mathcal{J})$ 4 $\mathcal{O}' \leftarrow \mathcal{O}_r$ 5 for $\beta \in H$ do 6 $\mathcal{O}' \leftarrow \mathcal{O}' \setminus \{\beta\}$ 7 return \mathcal{O}' </pre>
--

This procedure is illustrated by the following example:

Example 4.6. Let us consider an adapted version of the knowledge base of [Example 2.17](#):

$$\mathcal{O} = \{\text{alphaZero} : \text{ChessPlayer}, \text{alphaZero} : \text{GoPlayer}, \text{ChessPlayer} \sqcup \text{GoPlayer} \sqsubseteq \text{Person}, \text{alphaZero} : \text{ComputerProgram}\}.$$

Assume that $\mathcal{O}_s = \{\text{alphaZero} : \text{ComputerProgram}\}$ (of course, $\mathcal{O}_r = \mathcal{O} \setminus \mathcal{O}_s$). Let α be the sentence $\text{alphaZero} : \text{Person}$. In order to compute a repair of \mathcal{O} with respect to α , [Algorithm 1](#) starts by computing $\mathcal{J} := \text{Just}(\mathcal{O}, \alpha) = \{\{\text{alphaZero} : \text{ChessPlayer}, \text{ChessPlayer} \sqcup \text{GoPlayer} \sqsubseteq \text{Person}\}, \{\text{alphaZero} : \text{GoPlayer}, \text{ChessPlayer} \sqcup \text{GoPlayer} \sqsubseteq \text{Person}\}\}$. Then, it obtains a minimal hitting set of \mathcal{J} , which may be the set $H := \{\{\text{ChessPlayer} \sqcup \text{GoPlayer} \sqsubseteq \text{Person}\}\}$. Lastly, it returns the set obtained by removing from \mathcal{O}_r the elements of H , i.e. the set $\mathcal{O}' := \{\text{alphaZero} : \text{ChessPlayer}, \text{alphaZero} : \text{GoPlayer}\}$.

A special case of Ontology Repair is *ABox Repair*, where the TBox is fixed, i.e., the TBox is contained in \mathcal{O}_s :

Definition 4.7 (ABox Repair [LLR⁺10]). Let \mathcal{O} be an ontology, with TBox \mathcal{T} and ABox \mathcal{A} . An *ABox Repair* of \mathcal{O} is an inclusion-maximal subset \mathcal{A}' of \mathcal{A} such that the ontology \mathcal{O}' consisting of $\mathcal{T} \cup \mathcal{A}'$ is consistent.

It is easy to see that when $\mathcal{T} = \mathcal{O}_s$ and $\mathcal{A} = \mathcal{O}_r$, an ABox repair is an optimal repair according to [Definition 4.3](#).

In [Chapter 3](#), we have shown that contraction operations in classical Belief Revision are too restrictive for belief bases because of the inclusion postulate, and we analysed pseudo-contraction operations — a generalisation of contraction that satisfies logical inclusion rather than inclusion. Similarly, in Ontology Repair, classical repairs do not allow the inclusion of new sentences, and the same issue is present: sentences are either kept or removed altogether. In our [Example 4.6](#), the sentence $\text{ChessPlayer} \sqcup \text{GoPlayer} \sqsubseteq \text{Person}$ was discarded, but we might want to replace it with

a less constraining sentence that preserves some of the original information. A very similar idea to pseudo-contractions was introduced by Baader et al. [BKNP18] in Ontology Repair: in a *gentle repair*, one can either remove an axiom or substitute it with a weaker version, retaining part of the information it represented.

Definition 4.8 (Weakening [BKNP18]). A sentence α_1 is *weaker* than a sentence α_2 if $\text{Cn}(\{\alpha_1\}) \subset \text{Cn}(\{\alpha_2\})$.

Definition 4.9 (Gentle Repair [BKNP18]²). Let $\mathcal{O} = \langle \mathcal{O}_s, \mathcal{O}_r \rangle$ be an ontology and let α be a sentence entailed by \mathcal{O} but not by \mathcal{O}_s . An ontology \mathcal{O}' is a *gentle repair* of \mathcal{O} with respect to α if $\text{Cn}(\mathcal{O}_s \cup \mathcal{O}') \subseteq \text{Cn}(\mathcal{O}) \setminus \{\alpha\}$ and, for every $\varphi \in \mathcal{O}'$, either $\varphi \in \mathcal{O}_r$ or φ is weaker than ψ for some $\psi \in \mathcal{O}_r \setminus \mathcal{O}'$.

Algorithm 2 is very similar to Algorithm 1, but for every sentence in the hitting set H , a weaker sentence is used to replace it, thus giving us a gentle repair instead of a classical repair.

Algorithm 2: Gentle repair algorithm

<p>Input: An ontology $\mathcal{O} = \langle \mathcal{O}_s, \mathcal{O}_r \rangle$ and a formula α Output: A gentle repair \mathcal{O}' of \mathcal{O} w.r.t. α</p> <pre> 1 Function GentleRepair(\mathcal{O}, α) 2 $\mathcal{O}' \leftarrow \mathcal{O}_r$ 3 while $\alpha \in \text{Cn}(\mathcal{O}_s \cup \mathcal{O}')$ do 4 $\mathcal{J} \leftarrow \text{Justifications}(\langle \mathcal{O}_s, \mathcal{O}' \rangle, \alpha)$ 5 $H \leftarrow \text{MinimalHittingSet}(\mathcal{J})$ 6 for $\beta \in H$ do 7 $\beta' \leftarrow \text{GetWeakerSentence}(\mathcal{O}_s, \mathcal{J}, \alpha, \beta)$ 8 $\mathcal{O}' \leftarrow (\mathcal{O}' \setminus \{\beta\}) \cup \{\beta'\}$ 9 return \mathcal{O}' </pre>
--

In order to illustrate what is different in the outcome of this algorithm, we will use the same example that we presented to explain Algorithm 1.

Example 4.10. Consider again the problem discussed in Example 4.6. Starting with $\mathcal{O}' = \mathcal{O}_r$, Algorithm 2 computes \mathcal{J} and H exactly the way Algorithm 1 does. Then, instead of removing the sentence $\beta := \text{ChessPlayer} \sqcup \text{GoPlayer} \sqsubseteq \text{Person}$, Algorithm 2 tries to replace it with a weaker version, such as $\beta' := (\text{ChessPlayer} \sqcup \text{GoPlayer}) \sqcap \neg \text{ComputerProgram} \sqsubseteq \text{Person}$. This procedure is repeated until the set $\mathcal{O}_s \cup \mathcal{O}'$ fails to entail α . In our example, replacing β with β' is enough to prevent such entailment, and the algorithm stops, returning the repair $\{\mathcal{O}_r \setminus \{\beta\}\} \cup \{\beta'\}$.

A modified version of Algorithm 2 was proposed by Baader et al. [BKNP18] where instead of weakening each element of the minimal hitting set, only a single formula in each justification needs

²In [BKNP18], the concept of gentle repair has not been formally defined, only explained in intuitive terms. This is the definition which will be used here.

to be changed:

Algorithm 3: Modified gentle repair algorithm	
Input:	An ontology $\mathcal{O} = \langle \mathcal{O}_s, \mathcal{O}_r \rangle$ and a formula α
Output:	A gentle repair \mathcal{O}' of \mathcal{O} w.r.t. α
1	Function <i>ModifiedGentleRepair</i> (\mathcal{O}, α)
2	$\mathcal{O}' \leftarrow \mathcal{O}_r$
3	while $\alpha \in \text{Cn}(\mathcal{O}_s \cup \mathcal{O}')$ do
4	$J \leftarrow \text{OneJustification}(\langle \mathcal{O}_s, \mathcal{O}' \rangle, \alpha)$
5	$\beta' \leftarrow \text{GetWeakerSentence}(\mathcal{O}_s, \{J\}, \alpha, \beta)$
6	$\mathcal{O}' \leftarrow (\mathcal{O}' \setminus \{\beta\}) \cup \{\beta'\}$
7	return \mathcal{O}'

Baader et al. [BKNP18] remark that as the unmodified version requires the computation of minimal hitting sets, which is expensive, the modified version has an important advantage, even though both are prone to consume exponential time.

Algorithm 2 and Algorithm 3 require a function `GetWeakerSentence` which, given \mathcal{O}_s , \mathcal{J} , α and β , returns a sentence β' weaker than β such that $\alpha \notin \text{Cn}(\mathcal{O}_s \cup (J \setminus \{\beta\}) \cup \{\beta'\})$ for every $J \in \mathcal{J}$ such that $\beta \in J$. Such a β' always exists: a tautology satisfies the requirements. However, replacing a sentence with a tautology is logically equivalent to removing it, which means that a classical repair is obtained if this function only returns tautologies. Algorithm 3 needs a function `OneJustification`(\mathcal{O}, α) that computes an element J of $\text{Just}(\mathcal{O}, \alpha)$. Both algorithms are guaranteed to stop after a number of steps that grows at most exponentially in the size of the refutable part:

Theorem 4.11. [BKNP18] *Algorithm 2 and Algorithm 3 always stop after a finite number of iterations (of lines 3–8 and 3–6, respectively), which is at most exponential in $|\mathcal{O}_r|$, regardless of the DL of $\mathcal{O} = \langle \mathcal{O}_s, \mathcal{O}_r \rangle$.*

4.2 Correspondence between Belief Revision and Repairs in Description Logics

In this section, we will analyse the close relationship between the concepts and constructions presented for Ontology Repair and Belief Revision. Also, we propose a unified notation for the operations.

We start by giving two definitions that generalise several concepts in the literature.

Definition 4.12 (Maximal Non-Implying Subsets [MGSW19]). Let B be a knowledge base, α a sentence, and Φ a set of static sentences (i.e. which should be preserved in any operation). The set of *maximal α -non-implying subsets of B with respect to Φ* , denoted by $\text{MaxNon}(B, \alpha, \Phi)$, is such that $X \in \text{MaxNon}(B, \alpha, \Phi)$ if and only if $X \subseteq B$, $\alpha \notin \text{Cn}(\Phi \cup X)$, and there is no Y such that $X \subset Y \subseteq B$ and $\alpha \notin \text{Cn}(\Phi \cup Y)$.

For brevity, we shall omit the last argument of MaxNon whenever it is empty: $\text{MaxNon}(B, \alpha) = \text{MaxNon}(B, \alpha, \emptyset)$.

Remark 4.13 ([MGSW19]). *If $\Phi \subseteq B$, then the maximal α -non-implying subsets of B with respect to Φ contain all of the elements of Φ , i.e., $X \supseteq \Phi$ for every $X \in \text{MaxNon}(B, \alpha, \Phi)$.*

Proof. If there is some $X \in \text{MaxNon}(B, \alpha, \Phi)$ such that $X \not\supseteq \Phi$, then the set $Y = X \cup \Phi$ is such that $X \subset Y \subseteq B$, and since $\Phi \cup Y = \Phi \cup X$, we have that $\alpha \notin \text{Cn}(\Phi \cup Y) = \text{Cn}(\Phi \cup X)$, violating the definition of MaxNon . ■

Definition 4.12 corresponds to Definition 2.1 if $\Phi = \emptyset$, i.e., $\text{MaxNon}(B, \alpha) = \text{Rem}[B, \alpha]$.

Definition 4.14 (Minimal Implying Subsets [MGSW19]). Let B be a knowledge base, α a sentence, and Φ a set of static sentences. The set of *minimal α -implying subsets of B with respect to Φ* , denoted by $\text{MinImp}(B, \alpha, \Phi)$, is such that $X \in \text{MinImp}(B, \alpha, \Phi)$ if and only if $X \subseteq B$, $\alpha \in \text{Cn}(\Phi \cup X)$, and there is no $Y \subset X$ such that $\alpha \in \text{Cn}(\Phi \cup Y)$.

As in the previous definition, the last argument will be omitted if empty: $\text{MinImp}(B, \alpha) = \text{MinImp}(B, \alpha, \emptyset)$.

Remark 4.15 ([MGSW19]). *The minimal α -implying subsets of B with respect to Φ do not contain elements of Φ , i.e., $X \cap \Phi = \emptyset$ for every $X \in \text{MinImp}(B, \alpha, \Phi)$.*

Proof. If there is some $X \in \text{MinImp}(B, \alpha, \Phi)$ such that $X \cap \Phi \neq \emptyset$, then the set $Y = X \setminus \Phi$ is such that $Y \subset X$, and since $\Phi \cup Y = \Phi \cup X$, we have that $\alpha \in \text{Cn}(\Phi \cup Y) = \text{Cn}(\Phi \cup X)$, which contradicts the definition of MinImp . ■

If $\Phi = \emptyset$, Definition 4.14 corresponds to Definition 2.14, i.e., $\text{MinImp}(B, \alpha) = \text{Ker}[B, \alpha]$. Definition 4.14 is also closely related to Definition 4.4: $\text{MinImp}(B, \alpha, \Phi) = \text{Just}(\langle \Phi, B \setminus \Phi \rangle, \alpha)$, or conversely, $\text{Just}(\langle \mathcal{O}_s, \mathcal{O}_r \rangle, \alpha) = \text{MinImp}(\mathcal{O}_s \cup \mathcal{O}_r, \alpha, \mathcal{O}_s)$. Our definitions of MaxNon and MinImp also correspond, respectively, to the sets of MaNAs (maximal non-axiom sets) and MinAs (minimal axiom sets) in the literature [BP07]. In [FH10], the set of MISs (minimal inconsistent sets) of an inconsistent ontology \mathcal{O} corresponds to $\text{MaxNon}(\mathcal{O}, \perp)$. Furthermore, in the mathematical logic literature, the definition of argument [BH01] is related to our definition of minimal implying subset³.

Since we usually represent sentences by lowercase Greek letters, we propose to replace the classical Belief Revision notations γ and σ by \mathbf{g} and \mathbf{f} , respectively, to represent selection and incision functions (Definition 2.2 and Definition 2.15), as we have already done in this text.

The usual notations for partial meet contraction and kernel contraction overlap, making it impossible to distinguish between the two without the context (i.e. $B -_{\delta} \alpha$ could be either construction depending on what δ is). We propose a clearer notation for these constructions, which we have used throughout this text: $\text{pmc}_{\mathbf{g}}(B, \alpha)$ for partial meet contraction and $\text{kc}_{\mathbf{f}}(B, \alpha)$ for kernel contraction. Similarly, a generic contraction operation will be represented by $\text{c}(B, \alpha)$. Since the constructions for pseudo-contractions are the same, we use the same symbols (i.e. we will not add a ‘p’ for ‘pseudo’), and we add a superscript Cn^* whenever the operation closes the belief base under Cn^* before applying the partial meet or kernel constructions. A double star (Cn^{**}) is used for consequence relations in two-place operations, where the notion of consequence may be defined differently for each sentence.

We can now proceed to analyse the relation between the definitions and operations of Belief Revision and Ontology Repair.

Let $B \subseteq \mathcal{L}$ and $\alpha \in \mathcal{L}$. The following two properties follow straightly from Definition 2.14 and Definition 4.4.

Proposition 4.16 (Kernel \sim Justification [MGSW19]). *If $\alpha \in \text{Cn}(B)$, then a set X is an α -kernel of B with respect to α if and only if X is a justification for α in $\langle \emptyset, B \rangle$.*

In our notation, the set of all such sets X is denoted by $\text{MinImp}(B, \alpha)$, which unifies the concepts of the following proposition:

Proposition 4.17 (Kernel set \sim Set of all justifications [MGSW19]). *If $\alpha \in \text{Cn}(B)$, then $\text{Ker}[B, \alpha] = \text{Just}(\langle \emptyset, B \rangle, \alpha)$.*

A classical repair (Definition 4.2) can be seen as a contraction operation that satisfies two of Hansson’s postulates for base contraction.

³Assuming an implicit finite set of formulae Δ , called *database*, an *argument* is a pair $\langle \Phi, \alpha \rangle$ such that X is a minimal α -implying subset of Δ .

Proposition 4.18 (Classical Repair \implies Postulates for base contraction [MGSW19]). *Let Rep be an operation that yields a classical repair. Define the operation c_{Rep} as*

$$c_{\text{Rep}}(B, \alpha) = \begin{cases} \text{Rep}(\langle \emptyset, B \rangle, \alpha), & \text{if } B \models \alpha; \\ B, & \text{otherwise.} \end{cases}$$

Then, c_{Rep} satisfies success and inclusion.

The following proposition, which is an immediate consequence of the upper bound property (Proposition 2.12), will be useful to show the connection between partial meet base contraction and classical repairs.

Proposition 4.19 (Existence of α -remainder preserving \mathcal{O}_s [MGSW19]). *Let $\mathcal{O} = \langle \mathcal{O}_s, \mathcal{O}_r \rangle$ be an ontology and α be a sentence entailed by \mathcal{O} but not by \mathcal{O}_s . Then, there is at least one α -remainder X of $\mathcal{O}_s \cup \mathcal{O}_r$ such that $\mathcal{O}_s \subseteq X$.*

Now we can show that partial meet base contractions that include the static part of the ontology yield classical repairs.

Theorem 4.20 (Partial meet base contraction \implies Classical repair [MGSW19]). *Under the conditions of Proposition 4.19, if the selection function \mathbf{g} is such that $\mathcal{O}_s \subseteq X$ for every $X \in \mathbf{g}(\text{Rem}[\mathcal{O}, \alpha])$, then the operation $\text{Rep}_{\mathbf{g}}$ defined as*

$$\text{Rep}_{\mathbf{g}}(\mathcal{O}, \alpha) = \text{pmc}_{\mathbf{g}}(\mathcal{O}, \alpha) \setminus \mathcal{O}_s$$

yields a classical repair.

Proof. Let $\mathcal{O}' = \text{Rep}_{\mathbf{g}}(\mathcal{O}, \alpha)$. Since \mathbf{g} only selects α -remainders including \mathcal{O}_s , we have that $\mathcal{O}_s \subseteq \text{pmc}_{\mathbf{g}}(\mathcal{O}, \alpha)$, which implies that $\mathcal{O}_s \cup \mathcal{O}' = \text{pmc}_{\mathbf{g}}(\mathcal{O}, \alpha)$. Hence, from the inclusion postulate, we have that $\mathcal{O}_s \cup \mathcal{O}' \subseteq \mathcal{O}$, and monotonicity of Cn gives $\text{Cn}(\mathcal{O}_s \cup \mathcal{O}') \subseteq \text{Cn}(\mathcal{O})$. This is sufficient to show that the result of $\text{Rep}_{\mathbf{g}}$ is a repair. From the inclusion postulate, we have that $\text{pmc}_{\mathbf{g}}(\mathcal{O}, \alpha) \subseteq \mathcal{O}$, which proves that $\mathcal{O}' = \text{pmc}_{\mathbf{g}}(\mathcal{O}, \alpha) \setminus \mathcal{O}_s \subseteq \mathcal{O}_r$. Therefore, $\text{Rep}_{\mathbf{g}}$ yields a classical repair. ■

We summarise the relationships between essential concepts in both Belief Change and Ontology Repair using the diagram in Figure 4.1. In the diagram, we represent each concept with an ellipse and the areas by solid rectangles. Moreover, in the Belief Change area, we separate the concepts that constitute the partial meet approach from those that are part of the kernel approach as discussed earlier. We can note how most of the concepts have direct connections; the only exceptions are MIPS (set of Minimal Incoherence-Preserving Sub-TBoxes) and MUPS (set of Minimal Unsatisfiability-Preserving Sub-TBoxes) [SC03], which are variants of the usual justifications, and as such cannot be mapped directly to our definition of MinImp. More specifically, if \mathcal{T} is a TBox and A a concept name, then the set of MUPSes of A in \mathcal{T} is equivalent to $\text{MinImp}(\mathcal{T}, A \sqsubseteq \perp)$; and the set of all MIPSes in \mathcal{T} is given by $\text{MIPS}(\mathcal{T}) = \min_{\subseteq} \left\{ \bigcup_{A \in \mathcal{N}_{\mathcal{C}}(\mathcal{T})} \text{MinImp}(\mathcal{T}, A \sqsubseteq \perp) \right\}$, where $\mathcal{N}_{\mathcal{C}}(\mathcal{T})$ is the set of concept names in \mathcal{T} .

We can now show the relationship between pseudo-contractions and gentle repairs.

Proposition 4.21 (Gentle Repair \implies Pseudo-contraction). *Let GRep be an operation that yields a gentle repair. Define the operation $c_{(\text{GRep})}$ as*

$$c_{(\text{GRep})}(B, \alpha) = \begin{cases} \text{GRep}(\langle \emptyset, B \rangle, \alpha), & \text{if } B \models \alpha; \\ B, & \text{otherwise.} \end{cases}$$

Then, $c_{(\text{GRep})}$ is a pseudo-contraction operation.

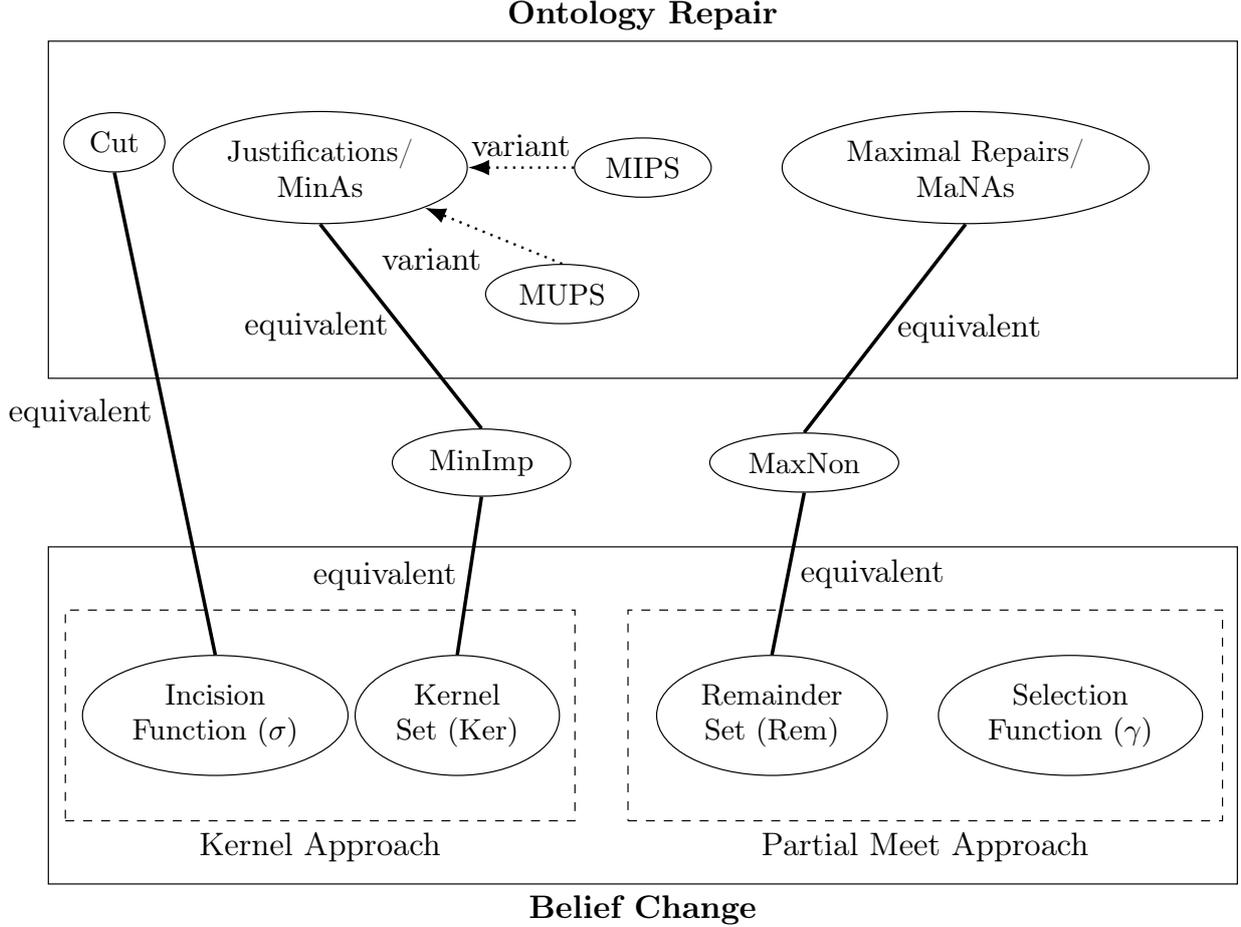


Figure 4.1: Relationship between concepts in Belief Change and Ontology Repair.

The result above is similar to [Proposition 4.18](#) and follows from [Definition 4.9](#), which guarantees that $c_{(\text{GRep})}$ satisfies success and logical inclusion.

For the other direction (pseudo-contractions as gentle repairs), we will introduce the notions of general partial meet pseudo-contractions and general kernel pseudo-contractions.⁴ Pseudo-contractions allow the result to contain some weakened versions of formulae that were originally in the belief base. This can be achieved by applying a partial meet or kernel operation on a ‘weak closure’ of the belief base (the original set plus *some* of its classical consequences) [[SMRW18](#)]. However, as this weak closure does not depend on the sentence that is being contracted, we are not able to add only weakenings of formulae that would be removed. General (partial meet and kernel) pseudo-contractions employ a consequence operator that depends on both the set of beliefs and the input sentence. Before defining them, we need the following concepts:

Definition 4.22 (Extension of a selection function [[RW08b](#), adapted]). Let g be a selection function for B , and let $B \subseteq B^* \subseteq \mathcal{L}$. We say that a selection function g' for B^* is an *extension* of g to B^* if g' is such that for every $\alpha \in \mathcal{L}$ and $X \in g(\text{MaxNon}(B, \alpha))$ there is a $Y \in g'(\text{MaxNon}(B^*, \alpha))$ such that $X \subseteq Y$.

Definition 4.23 (Extension of an incision function). Let f be an incision function for a set of sentences B , and let $B \subseteq B^* \subseteq \mathcal{L}$. The incision function f' for B^* is an *extension* of f for B^* if $f'(\text{MinImp}(B^*, \alpha)) \supseteq f(\text{MinImp}(B, \alpha))$ for all sentences α .

Now we can define the general partial meet pseudo-contraction⁵:

⁴Some of the definitions and results presented between this point and the end of the chapter – namely, the ones that involve kernel constructions – have yet to be published.

⁵In [[MGSW19](#)], this operation was referred to as ‘two-place partial meet pseudo-contraction’, which in [[SMRW18](#)]

Definition 4.24 (General partial meet pseudo-contraction [RW08b, SMRW18]). Let $\alpha \in \mathcal{L}$, $B \subseteq \mathcal{L}$, Cn' be a consequence relation, and \mathbf{g} be a selection function for B . Let us define $\text{Cn}^{**}(B, \alpha) := \text{Cn}'(B \setminus \bigcap \mathbf{g}(\text{MaxNon}(B, \alpha))) \cup B$, and let \mathbf{g}' be an extension of \mathbf{g} to $\text{Cn}^{**}(B, \alpha)$. The *general partial meet pseudo-contraction* of B by α , denoted by $\text{gpmc}_{\mathbf{g}, \mathbf{g}'}^{\text{Cn}^{**}}(B, \alpha)$, is defined as the two-place Cn^{**} partial meet pseudo-contraction of B by α (using \mathbf{g}'), i.e. the set $\bigcap \mathbf{g}'(\text{MaxNon}(\text{Cn}^{**}(B, \alpha), \alpha))$.

Notice that $B \subseteq \text{Cn}^{**}(B, \alpha)$ for all α , and so \mathbf{g}' can be an extension of \mathbf{g} . The construction above was proposed by Ribeiro and Wassermann [RW08b] (and generalised by Santos et al. [SMRW18]) as a way to weaken sentences in belief base pseudo-contractions, instead of removing them.

A similar idea is used to define the general kernel pseudo-contraction:

Definition 4.25 (General kernel pseudo-contraction). Let $\alpha \in \mathcal{L}$, $B \subseteq \mathcal{L}$, Cn' be a consequence relation, and \mathbf{f} be an incision function for B . Let us define $\text{Cn}^{**}(B, \alpha) := B \cup \text{Cn}'(\mathbf{f}(\text{MinImp}(B, \alpha)))$, and let \mathbf{f}' be an extension of \mathbf{f} to $\text{Cn}^{**}(B, \alpha)$. The *general kernel pseudo-contraction* of B by α , denoted by $\text{gkc}_{\mathbf{f}, \mathbf{f}'}^{\text{Cn}^{**}}(B, \alpha)$, is defined as the two-place Cn^{**} kernel pseudo-contraction of B by α (using \mathbf{f}'), i.e. the set $\text{Cn}^{**}(B, \alpha) \setminus \mathbf{f}'(\text{MinImp}(\text{Cn}^{**}(B, \alpha), \alpha))$.

Consider the following properties for selection and incision functions:

Definition 4.26 (A -inclusion [MGSW19]). Let $B \subseteq \mathcal{L}$, and let $A \subseteq B$. A selection function \mathbf{g} for B satisfies A -inclusion if, for all $\alpha \notin \text{Cn}(A)$, $A \subseteq X$ for every $X \in \mathbf{g}(\text{MaxNon}(B, \alpha))$.

Definition 4.27 (A -exclusion). Let $B \subseteq \mathcal{L}$, and let $A \subseteq B$. An incision function \mathbf{f} for B satisfies A -exclusion if $A \cap \mathbf{f}(\text{MinImp}(B, \alpha)) = \emptyset$ for all $\alpha \notin \text{Cn}(A)$.

Intuitively, a selection function (for B) that satisfies A -inclusion only selects α -remainders that preserve A , unless α itself is entailed by A , in which case A cannot be a subset of any α -remainder; similarly, an incision function (for B) that satisfies A -exclusion only selects sentences that are not in A , preserving A in the operation, unless α is entailed by A , in which case it is impossible to have an incision function that does not contain elements of A . We will now prove another result.

Lemma 4.28. *Consider a general partial meet pseudo-contraction as in Definition 4.24. Then, for all $\varphi \in B \setminus \bigcap \mathbf{g}(\text{MaxNon}(B, \alpha))$, there is an $X \in \mathbf{g}'(\text{MaxNon}(\text{Cn}^{**}(B, \alpha), \alpha))$ such that $\varphi \notin X$.*

Proof. Assume that $\varphi \in B \setminus \bigcap \mathbf{g}(\text{MaxNon}(B, \alpha))$. We need to show that there is some $X \in \mathbf{g}'(\text{MaxNon}(\text{Cn}^{**}(B, \alpha), \alpha))$ such that $\varphi \notin X$. Since $\varphi \notin \bigcap \mathbf{g}(\text{MaxNon}(B, \alpha))$, there is a $Y \in \mathbf{g}(\text{MaxNon}(B, \alpha))$ such that $\varphi \notin Y$. We have $Y \subseteq B$, $\alpha \notin \text{Cn}(Y)$ and for any $Y' \subseteq B$ such that $Y \subset Y'$, $\alpha \in \text{Cn}(Y')$. So, since \mathbf{g}' is an extension of \mathbf{g} to $\text{Cn}^{**}(B, \alpha)$, there is an $X \in \mathbf{g}'(\text{MaxNon}(\text{Cn}^{**}(B, \alpha), \alpha))$ such that $Y \subseteq X$. And since $\varphi \notin Y$, $\varphi \in B$ and for any $Y' \subseteq B$ such that $Y \subset Y'$ we have $\alpha \in \text{Cn}(Y')$, we conclude that $\varphi \notin X$ (otherwise, X would be a such Y' and α would be in $\text{Cn}(X)$). ■

If a consequence operator returns only sentences that are in the given set or are weaker than some of its sentences, then we say it is *strictly weakening*:

Definition 4.29 (Strictly weakening operator [MGSW19]). A consequence operator Con is *strictly weakening* if, for every $\varphi \in \mathcal{L}$ and every $B \subseteq \mathcal{L}$, $\varphi \in \text{Con}(B)$ if and only if $\varphi \in B$ or $\text{Con}(\{\varphi\}) \subset \text{Con}(\{\psi\})$ for some $\psi \in B$.

Now we can show under which conditions a general (partial meet or kernel) pseudo-contraction yields a gentle repair. We will need the following lemma:

Lemma 4.30. *Let c be a contraction operation for $B \subseteq \mathcal{L}$. Let $c^{\text{Cn}^{**}}$ be a pseudo-contraction operation such that $c^{\text{Cn}^{**}}(B, \beta) \subseteq \text{Cn}^{**}(B)$, where $(B \setminus c(B, \beta)) \cap c^{\text{Cn}^{**}}(B, \beta) = \emptyset$ and $\text{Cn}^{**}(B) := B \cup \text{Cn}'(B \setminus c(B, \beta))$ for all sentences β and the consequence relation Cn' is monotonic, subclassical and strictly weakening. If $\mathcal{O} := \langle \mathcal{O}_s, \mathcal{O}_r \rangle$ is an ontology such that $\mathcal{O}_s \subseteq c(\mathcal{O}, \beta) \cap c^{\text{Cn}^{**}}(\mathcal{O}, \beta)$ for all sentences β and α is a sentence such that $\alpha \notin \text{Cn}(\mathcal{O}_s)$, then the set $\mathcal{O}' := c^{\text{Cn}^{**}}(\mathcal{O}, \alpha) \setminus \mathcal{O}_s$ is a gentle repair of \mathcal{O} with respect to α .*

(and in this text) refers to a more general type of operations (see Definition 3.22).

Proof. We will start by proving that $\text{Cn}(\mathcal{O}_s \cup \mathcal{O}') \subseteq \text{Cn}(\mathcal{O}) \setminus \{\alpha\}$. By subclassicality of Cn' , it follows that $c^{\text{Cn}^{**}}(\mathcal{O}, \alpha) \subseteq \mathcal{O} \cup \text{Cn}(\mathcal{O} \setminus c(\mathcal{O}, \alpha))$, and by monotonicity, inclusion and idempotence of Cn we get $\text{Cn}(c^{\text{Cn}^{**}}(\mathcal{O}, \alpha)) \subseteq \text{Cn}(\mathcal{O})$. Now, since $\mathcal{O}_s \subseteq c^{\text{Cn}^{**}}(\mathcal{O}, \alpha)$, we have $\mathcal{O}_s \cup \mathcal{O}' = \mathcal{O}_s \cup (c^{\text{Cn}^{**}}(\mathcal{O}, \alpha) \setminus \mathcal{O}_s) = c^{\text{Cn}^{**}}(\mathcal{O}, \alpha)$, so $\text{Cn}(\mathcal{O}_s \cup \mathcal{O}') \subseteq \text{Cn}(\mathcal{O})$. To show that $\alpha \notin \text{Cn}(\mathcal{O}_s \cup \mathcal{O}')$, since we have just shown that $\mathcal{O}_s \cup \mathcal{O}' = c^{\text{Cn}^{**}}(\mathcal{O}, \alpha)$, from success of $c^{\text{Cn}^{**}}$ we have that $\alpha \notin \text{Cn}(c^{\text{Cn}^{**}}(\mathcal{O}, \alpha))$, and the result follows. \blacksquare

Now we have to prove that, for all $\varphi \in \mathcal{O}'$, either $\varphi \in \mathcal{O}_r$ or $\text{Cn}(\{\varphi\}) \subset \text{Cn}(\{\psi\})$ for some $\psi \in \mathcal{O}_r \setminus \mathcal{O}'$. Take some $\varphi \in \mathcal{O}'$. If $\varphi \in \mathcal{O}_r$, this part of the proof is done. Assume that $\varphi \notin \mathcal{O}_r$. As $\varphi \in \mathcal{O}'$, we have $\varphi \in c^{\text{Cn}^{**}}(\mathcal{O}, \alpha)$, but $c^{\text{Cn}^{**}}(\mathcal{O}, \alpha) \subseteq \text{Cn}^{**}(\mathcal{O}, \alpha) = \text{Cn}'(\mathcal{O} \setminus c(\mathcal{O}, \alpha)) \cup \mathcal{O}$. It was assumed that $\varphi \notin \mathcal{O}_r$, so $\varphi \notin \mathcal{O}$ (because $\varphi \in \mathcal{O}'$ implies that $\varphi \notin \mathcal{O}_s$), and thus φ must be in $\text{Cn}'(\mathcal{O} \setminus c(\mathcal{O}, \alpha))$. Since Cn' is strictly weakening, either $\varphi \in \mathcal{O} \setminus c(\mathcal{O}, \alpha)$ or there is a $\psi \in \mathcal{O} \setminus c(\mathcal{O}, \alpha)$ such that $\text{Cn}(\{\varphi\}) \subset \text{Cn}(\{\psi\})$. The first case is not possible: since $\mathcal{O}_s \subseteq c(\mathcal{O}, \alpha)$, we have $\mathcal{O} \setminus c(\mathcal{O}, \alpha) \subseteq \mathcal{O}_r$; hence, φ cannot be in $\mathcal{O} \setminus c(\mathcal{O}, \alpha)$ because we assumed that $\varphi \notin \mathcal{O}_r$. Take some $\psi \in \mathcal{O} \setminus c(\mathcal{O}, \alpha)$ such that $\text{Cn}(\{\varphi\}) \subset \text{Cn}(\{\psi\})$. Since $\mathcal{O} \setminus c(\mathcal{O}, \alpha) \subseteq \mathcal{O}_r$, we know that $\psi \in \mathcal{O}_r$. Now it is left to show that $\psi \notin \mathcal{O}'$, i.e., $\psi \notin c^{\text{Cn}^{**}}(\mathcal{O}, \alpha)$ or $\psi \in \mathcal{O}_s$. Since $\psi \in \mathcal{O}_r$ and \mathcal{O}_s and \mathcal{O}_r are assumed to be disjoint, $\psi \notin \mathcal{O}_s$. So we only have to show that $\psi \notin c^{\text{Cn}^{**}}(\mathcal{O}, \alpha)$. We know that $\psi \in \mathcal{O} \setminus c(\mathcal{O}, \alpha)$, and the fact that $(\mathcal{O} \setminus c(\mathcal{O}, \alpha)) \cap c^{\text{Cn}^{**}}(\mathcal{O}, \alpha) = \emptyset$ implies that ψ is not in $c^{\text{Cn}^{**}}(\mathcal{O}, \alpha)$, finishing the proof. \blacksquare

The following results show that, under certain conditions, a general partial meet pseudo-contraction and a general kernel pseudo-contraction produce gentle repairs.

Theorem 4.31 (General Partial Meet Pseudo-Contraction \implies Gentle Repair [MGSW19, adapted]). *Let $\text{gpmc}_{\mathbf{g}, \mathbf{g}'}^{\text{Cn}^{**}}$ and Cn^{**} be as in Definition 4.24, Cn^{**} based on a consequence relation Cn' that satisfies subclassicality, \mathbf{g} and \mathbf{g}' satisfy \mathcal{O}_s -inclusion, Cn' be monotonic and strictly weakening, and $\mathcal{O} = \langle \mathcal{O}_s, \mathcal{O}_r \rangle$. If $\alpha \notin \text{Cn}(\mathcal{O}_s)$, then $\mathcal{O}' := \text{gpmc}_{\mathbf{g}, \mathbf{g}'}^{\text{Cn}^{**}}(\mathcal{O}, \alpha) \setminus \mathcal{O}_s$ is a gentle repair of \mathcal{O} w.r.t. α .*

Proof. The result follows from Lemma 4.30 by taking $\text{pmc}_{\mathbf{g}}$ as c and $\text{gpmc}_{\mathbf{g}, \mathbf{g}'}^{\text{Cn}^{**}}$ as $c^{\text{Cn}^{**}}$. Lemma 4.28 shows that the assumption $(B \setminus c(B, \beta)) \cap c^{\text{Cn}^{**}}(B, \beta) = \emptyset$ holds, and the assumption that $\mathcal{O}_s \subseteq c(\mathcal{O}, \beta) \cap c^{\text{Cn}^{**}}(\mathcal{O}, \beta)$ for all sentences β derives from \mathcal{O}_s -inclusion of \mathbf{g} and \mathbf{g}' . \blacksquare

Theorem 4.32 (General Kernel Pseudo-Contraction \implies Gentle Repair). *Let $\text{gkc}_{\mathbf{f}, \mathbf{f}'}^{\text{Cn}^{**}}$ and Cn^{**} be as in Definition 4.25, Cn^{**} based on a consequence relation Cn' that satisfies subclassicality, \mathbf{f} and \mathbf{f}' satisfy \mathcal{O}_s -exclusion, Cn' be monotonic and strictly weakening, and $\mathcal{O} = \langle \mathcal{O}_s, \mathcal{O}_r \rangle$. If $\alpha \notin \text{Cn}(\mathcal{O}_s)$, then $\mathcal{O}' := \text{gkc}_{\mathbf{f}, \mathbf{f}'}^{\text{Cn}^{**}}(\mathcal{O}, \alpha) \setminus \mathcal{O}_s$ is a gentle repair of \mathcal{O} w.r.t. α .*

Proof. The result follows from Lemma 4.30 by taking $\text{kc}_{\mathbf{f}}$ as c and $\text{gkc}_{\mathbf{f}, \mathbf{f}'}^{\text{Cn}^{**}}$ as $c^{\text{Cn}^{**}}$. The definition of extension of an incision function is enough to conclude that the assumption $(B \setminus c(B, \beta)) \cap c^{\text{Cn}^{**}}(B, \beta) = \emptyset$ holds, and the assumption that $\mathcal{O}_s \subseteq c(\mathcal{O}, \beta) \cap c^{\text{Cn}^{**}}(\mathcal{O}, \beta)$ for all sentences β derives from \mathcal{O}_s -exclusion of \mathbf{f} and \mathbf{f}' . \blacksquare

Chapter 5

Conclusion and future work

In this thesis, we have extended previous work on pseudo-contraction, which was initially proposed by Hansson [Han93a] and further studied in Ribeiro and Wassermann [RW08a] and Santos et al. [SMRW18]. While the previous studies used only partial meet operations, we have here defined an operation called Cn^* kernel pseudo-contraction, which is based on a kernel construction. We have defined an adapted set of postulates, which we have proved to be equivalent to the construction, characterising the operation by means of a representation theorem. We have extended previous code to implement the prototype of a pseudo-contraction plug-in for the ontology editor Protégé, allowing the computation of Cn^* (kernel and partial meet) pseudo-contractions in ontologies. We have also defined postulates for a new operation called pseudo-revision, which is to revision as pseudo-contraction is to contraction. Furthermore, we have proposed a unified notation for some Belief Revision and Ontology Repair concepts, which have several ideas in common, and we have proved that some of their definitions are equivalent under certain conditions.

Parts of the results appeared in [SMRW18] and [MGSW19].

Future work

A few ideas to expand the research in the areas covered by this text are given below:

- investigate the relationship between pseudo-contractions and the area of defeasible reasoning;
- evaluate the performance of pseudo-contractions in both artificial and real-world ontologies in order to compare the practical efficiency of the constructions;
- define pseudo-revision constructions that are equivalent to the postulates and do not depend on negation;
- adapt the pseudo-contraction constructions to support the removal of multiple sentences at once;
- analyse the relevance of syntax in a knowledge base from a philosophical perspective and its associations with hypothetical reasoning;
- examine the connection between pseudo-contractions and paraconsistent logics;
- create pseudo-contraction constructions that are based on ensconcement;
- explore families of consequence operators that are interesting for theoretical or practical purposes.

Bibliography

- [ABB⁺00] Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, J. Michael Cherry, Allan Peter Davis, Kara Dolinski, Selina S. Dwight, Janan T. Eppig, Midori A. Harris, David P. Hill, Laurie Issel-Tarver, Andrew Kasarskis, Suzanna Lewis, John C. Matese, Joel E. Richardson, Martin Ringwald, Gerald M. Rubin and Gavin Sherlock. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.
- [AGM85] Carlos Alchourrón, Peter Gärdenfors and David Makinson. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50(2):510–530, 1985.
- [AM81] Carlos Alchourrón and David Makinson. *New studies in Deontic Logic*, pages 125–148. Reidel Publishing Company, 1981.
- [Baa96] Franz Baader. Using automata theory for characterizing the semantics of terminological cycles. *Ann. Math. Artif. Intell.*, 18(2-4):175–219, 1996.
- [BBL05] Franz Baader, Sebastian Brandt and Carsten Lutz. Pushing the EL envelope. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 – August 5, 2005*, pages 364–369. Professional Book Center, 2005.
- [BH01] Philippe Besnard and Anthony Hunter. A logic-based theory of deductive arguments. *Artif. Intell.*, 128(1-2):203–235, 2001.
- [BHLS17] Franz Baader, Ian Horrocks, Carsten Lutz and Uli Sattler. *An Introduction to Description Logic*. Cambridge University Press, 2017.
- [BKM99] Franz Baader, Ralf Küsters and Ralf Molitor. Computing least common subsumers in description logics with existential restrictions. In Thomas Dean, editor, *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI 99, Stockholm, Sweden, July 31 - August 6, 1999. 2 Volumes, 1450 pages*, pages 96–103. Morgan Kaufmann, 1999.
- [BKNP18] Franz Baader, Francesco Kriegel, Adrian Nuradiansyah and Rafael Peñaloza. Making repairs in description logics more gentle. In Michael Thielscher, Francesca Toni and Frank Wolter, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018, Tempe, Arizona, 30 October — 2 November 2018.*, pages 319–328. AAAI Press, 2018.
- [BL84] Ronald J. Brachman and Hector J. Levesque. The tractability of subsumption in frame-based description languages. In Ronald J. Brachman, editor, *Proceedings of the National Conference on Artificial Intelligence. Austin, TX, USA, August 6-10, 1984.*, pages 34–37. AAAI Press, 1984.

- [BN03] Franz Baader and Werner Nutt. Basic description logics. In Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi and Peter F. Patel-Schneider, editors, *The Description Logic Handbook: Theory, Implementation, and Applications*, pages 43–95. Cambridge University Press, 2003.
- [BP07] Franz Baader and Rafael Peñaloza. Axiom pinpointing in general tableaux. In Nicola Olivetti, editor, *Automated Reasoning with Analytic Tableaux and Related Methods*, pages 11–27, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.
- [CW15] Raphael C obe and Renata Wassermann. Ontology repair through partial meet contraction. In Richard Booth, Giovanni Casini, Szymon Klarman, Gilles Richard and Ivan Jos e Varzinczak, editors, *Proceedings of the International Workshop on Defeasible and Ampliative Reasoning, DARE 2015, co-located with the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015), Buenos Aires, Argentina, July 27, 2015.*, volume 1423 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2015.
- [DM00] Francesco M. Donini and Fabio Massacci. EXPTIME tableaux for ALC. *Artificial Intelligence*, 124(1):87–138, 2000.
- [DQ15] Jianfeng Du and Guilin Qi. Tractable computation of representative ABox repairs in description logic ontologies. In Songmao Zhang, Martin Wirsing and Zili Zhang, editors, *Knowledge Science, Engineering and Management - 8th International Conference, KSEM 2015, Chongqing, China, October 28–30, 2015, Proceedings*, volume 9403 of *Lecture Notes in Computer Science*, pages 28–39. Springer, 2015.
- [FFK06] Marcelo A. Falappa, Eduardo L. Ferm e and Gabriele Kern-Isberner. On the logic of theory change: Relations between incision and selection functions. In Gerhard Brewka, Silvia Coradeschi, Anna Perini and Paolo Traverso, editors, *Proceedings of the 17th European Conference on Artificial Intelligence (ECAI 2006), Riva del Garda, Italy, August 29–September 1, 2006*, volume 141 of *Frontiers in Artificial Intelligence and Applications*, pages 402–406. IOS Press, 2006.
- [FH10] Jun Fang and Zhisheng Huang. Reasoning with inconsistent ontologies. *Tsinghua Science & Technology*, 15(6):687–691, 2010.
- [FMK⁺08] Giorgos Flouris, Dimitris Manakanatas, Haridimos Kondylakis, Dimitris Plexousakis and Grigoris Antoniou. Ontology change: Classification and survey. *Knowledge Engineering Review*, 23(2):117–152, 2008.
- [FPA05] 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.
- [G ar88] Peter G ardenfors. *Knowledge in Flux - Modeling the Dynamics of Epistemic States*. MIT Press, 1988.
- [Gui20] Ricardo Ferreira Guimar es. *Modularity in belief change of description logic bases*. PhD Thesis, Universidade de S o Paulo, S o Paulo, 2020.
- [GZB06] Christine Golbreich, Songmao Zhang and Olivier Bodenreider. The foundational model of anatomy in OWL: Experience and perspectives. *Journal of Web Semantics*, 4(3):181–195, 2006. Semantic Web for Life Sciences.
- [Han89] Sven Ove Hansson. New operators for theory change. *Theoria*, 55(2):114–132, 1989.
- [Han91] Sven Ove Hansson. Belief contraction without recovery. *Studia Logica*, 50(2):251–260, 1991.

- [Han93a] Sven Ove Hansson. Changes of disjunctively closed bases. *Journal of Logic, Language and Information*, 2(4):255–284, 1993.
- [Han93b] Sven Ove Hansson. Reversing the Levi identity. *Journal of Philosophical Logic*, 22:637–639, 1993.
- [Han94] Sven Ove Hansson. Kernel contraction. *The Journal of Symbolic Logic*, 59(3):845–859, 1994.
- [Han99] Sven Ove Hansson. *A textbook of belief dynamics - theory change and database updating*, volume 11 of *Applied logic series*. Kluwer, 1999.
- [HDG⁺06] Matthew Horridge, Nick Drummond, John Goodwin, Alan L. Rector, Robert Stevens and Hai Wang. The manchester OWL syntax. In Bernardo Cuenca Grau, Pascal Hitzler, Conor Shankey and Evan Wallace, editors, *Proceedings of the OWLED*06 Workshop on OWL: Experiences and Directions, Athens, Georgia, USA, November 10-11, 2006*, volume 216 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2006.
- [Hor11] Matthew Horridge. *Justification based explanation in ontologies*. PhD Thesis, University of Manchester, 2011.
- [HW02] Sven Ove Hansson and Renata Wassermann. Local change. *Studia Logica*, 70(1):49–76, 2002.
- [JHQ⁺09] Qiu Ji, Peter Haase, Guilin Qi, Pascal Hitzler and Steffen Stadtmüller. RaDON – repair and diagnosis in ontology networks. In *Proceedings of the 6th European Semantic Web Conference on The Semantic Web: Research and Applications, ESWC 2009 Heraklion*, pages 863–867, Berlin, Heidelberg, 2009. Springer-Verlag.
- [Kal06] Aditya Kalyanpur. *Debugging and Repair of OWL Ontologies*. PhD Thesis, University of Maryland at College Park, College Park, MD, USA, 2006.
- [KPHS07] Aditya Kalyanpur, Bijan Parsia, Matthew Horridge and Evren Sirin. Finding all justifications of OWL DL entailments. In *Proceedings of the 6th International Semantic Web Conference (ISWC 2007), Busan, Korea, November 11–15, 2007*, volume 4825 of *Lecture Notes in Computer Science*, pages 267–280. Springer, 2007.
- [LLR⁺10] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi and Domenico Fabio Savo. Inconsistency-tolerant semantics for description logics. In *Proceedings of the Fourth International Conference on Web Reasoning and Rule Systems (RR 2010)*, 2010.
- [LSPV08] Joey Sik Chun Lam, Derek H. Sleeman, Jeff Z. Pan and Wamberto Weber Vasconcelos. A fine-grained approach to resolving unsatisfiable ontologies. In Spaccapietra S., editor, *Journal on Data Semantics X*, volume 4900 of *Lecture Notes in Computer Science*, pages 62–95. Springer, 2008.
- [Mey01] Thomas Andreas Meyer. Basic infobase change. *Studia Logica*, 67(2):215–242, 2001.
- [MGH⁺12] Boris Motik, Bernardo Cuenca Grau, Ian Horrocks, Zhe Wu, Achille Fokoue and Carsten Lutz. OWL 2 Web Ontology Language profiles (second edition). <https://www.w3.org/TR/owl2-profiles/>, December 2012.
- [MGSW19] Vinícius Bitencourt Matos, Ricardo Ferreira Guimarães, Yuri David Santos and Renata Wassermann. Pseudo-contractions as gentle repairs. In Carsten Lutz, Uli Sattler, Cesare Tinelli, Anni-Yasmin Turhan and Frank Wolter, editors, *Description Logic, Theory Combination, and All That - Essays Dedicated to Franz Baader on the Occasion of His 60th Birthday*, volume 11560 of *Lecture Notes in Computer Science*, pages 385–403. Springer, 2019.

- [MvH04] Deborah Louise McGuinness and Frank van Harmelen. OWL Web Ontology Language Overview. <https://www.w3.org/TR/owl-features/>, February 2004.
- [Rei87] Raymond Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–95, April 1987.
- [Rib13] Márcio Moretto Ribeiro. *Belief Revision in Non-Classical Logics*. Springer London, London, 2013.
- [RW08a] Márcio Moretto Ribeiro and Renata Wassermann. The ontology reviser plug-in for protégé. In Frederico Luiz Gonçalves de Freitas, Heiner Stuckenschmidt, Helena Sofia Pinto, Andreia Malucelli and Óscar Corcho, editors, *Proceedings of the 3rd Workshop on Ontologies and their Applications, Salvador, Bahia, Brazil, October 26, 2008*, volume 427 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.
- [RW08b] Márcio Moretto Ribeiro and Renata Wassermann. Degrees of recovery and inclusion in belief base dynamics. In Maurice Pagnucco and Michael Thielscher, editors, *Proceedings of the Twelfth International Workshop on Non-Monotonic Reasoning*, pages 43–49, Sydney, September 2008.
- [RW09] Márcio Moretto Ribeiro and Renata Wassermann. Base revision for ontology debugging. *Journal of Logic and Computation*, 19(5):721–743, 2009.
- [San16] Yuri David Santos. *Pseudo-Contractions in Belief Revision*. Master’s thesis, Universidade de São Paulo, May 2016.
- [SC03] Stefan Schlobach and Ronald Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI 2003), Acapulco, Mexico, August 9–15, 2003*, pages 355–362. Morgan Kaufmann, 2003.
- [Sch05] Stefan Schlobach. Debugging and semantic clarification by pinpointing. In *Proceedings of the 2nd European Semantic Web Conference (ESWC 2005), Heraklion, Crete, Greece, May 29 – June 1, 2005*, volume 3532 of *Lecture Notes in Computer Science*, pages 226–240. Springer, 2005.
- [SMRW18] Yuri David Santos, Vinícius Bitencourt Matos, Márcio Moretto Ribeiro and Renata Wassermann. Partial meet pseudo-contractions. *International Journal of Approximate Reasoning*, 103:11–27, 2018.
- [SSS91] Manfred Schmidt-Schauß and Gert Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.
- [TCG⁺18] Nicolas Troquard, Roberto Confalonieri, Pietro Galliani, Rafael Peñaloza, Daniele Porello and Oliver Kutz. Repairing ontologies via axiom weakening. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1981–1988. AAAI Press, 2018.
- [The19] The Gene Ontology Consortium. The Gene Ontology Resource: 20 years and still GOing strong. *Nucleic Acids Research*, 47(D1):D330–D338, January 2019.
- [Was00] Renata Wassermann. An algorithm for belief revision. In Anthony G. Cohn, Fausto Giunchiglia and Bart Selman, editors, *KR 2000, Principles of Knowledge Representation and Reasoning Proceedings of the Seventh International Conference, Breckenridge, Colorado, USA, April 11–15, 2000*, pages 345–352. Morgan Kaufmann, 2000.
- [Was11] Renata Wassermann. On AGM for non-classical logics. *J. Philosophical Logic*, 40(2):271–294, 2011.