Bridging Belief Revision and Ontology Repair

moving closer to optimal repairs

Davy Alves de Souza

Thesis presented to the
Institute of Mathematics and Statistics
of the University of São Paulo
in partial fulfillment
of the requirements
for the degree of
Master of Science

Program: Computer Science
Advisor: Prof. Dr. Renata Wassermann

São Paulo
October, 2024

Bridging Belief Revision and Ontology Repair

moving closer to optimal repairs

Davy Alves de Souza

This is the original version of the thesis
prepared by candidate Davy Alves de Souza,
as submitted to the Examining Committee.

Ficha catalográfica elaborada com dados inseridos pelo(a) autor(a)
Biblioteca Carlos Benjamin de Lyra
Instituto de Matemática e Estatística
Universidade de São Paulo

```
Souza, Davy Alves de
   Bridging Belief Revision and Ontology Repair: moving
closer to optimal repairs / Davy Alves de Souza; orientadora,
Renata Wassermann. - São Paulo, 2024.
   68 p.: il.


   Dissertação (Mestrado) - Programa de Pós-Graduação
em Ciência da Computação / Instituto de Matemática e
Estatística / Universidade de São Paulo.
   Bibliografia
   Versão original


   1. Belief Revision. 2. Knowledge Representation. I.
Wassermann, Renata. II. Título.
```

Bibliotecárias do Serviço de Informação e Biblioteca
Carlos Benjamin de Lyra do IME-USP, responsáveis pela
estrutura de catalogação da publicação de acordo com a AACR2:
Maria Lúcia Ribeiro CRB-8/2766; Stela do Nascimento Madruga CRB 8/7534.

If you have knowledge, let others light their candles in it.
— Margaret Fuller

Acknowledgements

It’s the repetition of affirmations that leads to belief. And once that belief becomes a deep conviction, things begin to happen.
— Muhammad Ali

Acknowledgments Starting an MSc degree after a few years away from university might be challenging. Balancing work and studies in a healthy and productive way has almost always proven to be tiring and, at times, impossible. Nevertheless, I can say that none of this diminished my satisfaction when I made this decision. Returning to academic life revived a spirit of curiosity and the joy of discoveries, which the years as a professional had taken away from me.

Many people helped me, not only in making this decision but also in various moments and different ways throughout this period. First and foremost, I remember my family, who supported me during this time, even though the idea of traveling to another city and working overtime sometimes seemed wild. I am grateful to my brother, Nahim, who always encouraged me not to give up during this time.

One of the first challenges when starting a master’s program is finding someone who believes in you and is willing to guide you along this path. I want to thank my advisor, Renata Wassermann, who, even without knowing me, believed in me and helped me at every moment. The beginning of my MSc program coincided with a time of great fear and uncertainty for all of us, which was the year 2020. Renata helped me to keep going during this period and even took on the new challenge of advising a student despite the distance.

I also thank my research colleagues, whom I had the pleasure of meeting during these years and who, in some way, supported me. I thank my friends, who always supported me and told me this was the path to follow.

And finally, I thank God, who gives us love, faith, and hope to move forward and to whom all the glory belongs, now and forever.

Resumo

Davy Alves de Souza. **Conectando Revisão de Crenças e Reparo de Ontologias: *aproximando-se de reparos ótimos*.** Dissertação (Mestrado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2024.

Em Representação de Conhecimento, o conhecimento de um agente pode ser representado por um conjunto de crenças. A Revisão de Crenças pesquisa formas de lidar com possíveis mudanças em um conjunto de crenças. O modelo AGM descreve operações de expansão, revisão e contração, usadas como base para modelos de revisão. Alguns estudos demonstraram uma relação muito próxima entre a área de Revisão de Crenças e os estudos de Reparo de Ontologias. Nesta pesquisa, analisamos métodos de reparo que permitem a construção de reparos ótimos, comparando os seus resultados com os resultados de operações de contração. Além disso, apresentamos um método de saturação de bases de conhecimento, permitindo que operações de contração tenham resultados mais próximos aos reparos ótimos.

Palavras-chave: Revisão de Crenças. Reparo de Ontologias. Pseudo-contrações.

Abstract

Davy Alves de Souza. **Bridging Belief Revision and Ontology Repair: *moving closer to optimal repairs***. Thesis (Master's). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2024.

In Knowledge Representation, the knowledge of an agent can be represented by a belief set. Belief Revision explores methods to handle potential changes in a belief set. The AGM model describes operations of expansion, revision, and contraction, which serve as the foundation for revision models. Some studies demonstrated a close relationship between the field of Belief Revision and Ontology Repair. In this research, we analyze repair methods that are able to construct optimal repairs, comparing their results with those of contraction operations. Additionally, we present a method for saturating knowledge bases, allowing contraction operations to yield results that are closer to optimal repairs.

Keywords: Belief Revision. Ontology Repair. Pseudo-contractions.

List of Abbreviations

ABox	Assertional box
$\mathcal{AL}$	Attributive language
CQ	Conjunctive queries
DFS	Depth-first search
DL	Description logics
$\mathcal{EL}$	Existential language
$\mathcal{FL}$	Frame-based language
GCI	General concept inclusion
IQ	Instance queries
KB	Knowledge base
qABox	Quantified assertional box
QL	Query language
TBox	Terminological box

List of Figures

List of Tables

List of Programs

Contents

Appendixes

Introduction

Knowledge Representation (KR) is a field of artificial intelligence that aims to represent knowledge about the world, allowing reasoners to infer more information from it (Brachman and Levesque, 2004). Peppas (2008) says that knowledge is dynamic, and even a small change can significantly impact an entire knowledge base. Belief Revision explores methods to handle these changes.

The work of Alchourrón, Gärdenfors, and Makinson (1985), recognized as the starting point of Belief Revision, proposed a method that utilizes propositional logic to represent a belief set and operations to handle the changes in its beliefs. This model, which describes principles that characterize operations of expansion, revision, and contraction, was named the AGM model. From the AGM model, several generalizations were developed, including using other logics, for example, description logics (Franz Baader, Horrocks, *et al.*, 2017). Hansson (1993b) proposed a generalization of belief sets known as belief bases together with a new construction of contractions based on the notions of kernels and incision functions (Hansson, 1994).

Ontology Engineering studies how to model an ontology representing a domain in terms of concepts and roles. In building an ontology, the need for repair has proven to be critical in maintaining consistency. Ontology Repair (Kalyanpur, 2006; Keet, 2018), then, explores different methods to repair an ontology. Classical approaches significantly impact the knowledge base, leading to the unnecessary loss of information while repairing an ontology. Franz Baader, Kriegel, *et al.* (2018) present a strategy to make the repairs more gentle. While this method proved efficient in alleviating the impacts of a repair, it remained significantly dependent on the syntax of the axioms. In Franz Baader, Kriegel, *et al.* (2020), a method is introduced to remove unwanted assertions by using an extension of ABoxes called quantified ABoxes. In Franz Baader, Koopmann, Kriegel, and Nuradiansyah (2021), the authors presented an approach to finding optimal repairs by focusing on the assertions within the ontology while keeping the terminological part static.

Several studies have sought to map similarities between Belief Revision and Ontology Repair. Following the generalizations presented in Hansson (1993a), Santos *et al.* (2018) examined the properties of partial meet pseudo-contractions, defining a new construction of this operation that used a different consequence relation. Matos *et al.* (2019) extended the work of Santos *et al.* to the notions of kernel contractions and related the constructions of pseudo-contraction operations to gentle repair approaches. Franz Baader (2023) explored how functional approaches to construct optimal repairs align with pseudo-contraction operations under certain conditions.

The aim of this work is to explore the differences between optimal repairs and the outcomes of pseudo-contraction operations. We also present a method for saturating the knowledge base, which allows us to bring the results of these two approaches closer together. Chapter 1 introduces the notions of Belief Revision, the generalization for belief bases, and a detailed coverage of pseudo-contraction operations. Chapter 2 introduces the foundations of Description Logics, focusing on $\mathcal{EL}$. It also explains the characteristics of quantified ABoxes, which is crucial to understanding the functional approaches of repairs. Chapter 3 presents an overview of classical repair approaches as well as the intuition behind functional approaches, which lead us to optimal repairs. This chapter also presents examples of optimized repairs and some similarities between optimal repairs and pseudo-contraction operations. Finally, Chapter 4 discusses some differences between optimal repairs and pseudo-contraction operations while proposing a method to expand knowledge bases so the results of contractions come closer to optimal repairs.

Chapter 1

Belief Revision

Belief Revision is a field of Knowledge Representation that aims to understand the dynamics of epistemic states of an agent. An *epistemic state* of an agent represents all of their beliefs in a given moment (Ribeiro, 2010). In other words, Belief Revision seeks to comprehend the agent's behavior when faced with changes in its beliefs (Ribeiro, 2010; Matos, 2021).

The seminal contribution to Belief Revision was made by Alchourrón, Gärdenfors, *et al.* (1985), who proposed a method for representing the agent's epistemic state alongside a set of operations to manage the changes in its beliefs when receiving new information. The guiding principle behind this method is the *principle of minimal change*, which suggests that a rational agent should adjust its beliefs as little as possible in order to accommodate the new information (Peppas, 2008). Due to the authors' names, this model became known as the AGM framework.

1.1 Operations

In the AGM model, *beliefs* are represented by formulas of a propositional language, and the epistemic states of an agent are represented by logically closed sets of propositions, known as *belief sets* (Alchourrón, Gärdenfors, and Makinson, 1985). This model defines three operations that can be performed on belief sets: contraction, expansion, and revision.

1.1.1 Expansion

An *expansion* operation involves adding new information to the belief set, which could lead to an inconsistent belief state. If we have a belief set K and a belief φ, the resulting expansion $K + \varphi$ adds the new belief to the initial belief set and computes the logical consequences of the resulting set:

$$K + \varphi = \mathsf{Cn}(K \cup \{\varphi\}),$$

where Cn is a consequence operator and $K \subseteq K + \varphi$. Essentially, the belief set is expanded with the new belief and all its consequences. As the operation does not modify existing beliefs, it may introduce an inconsistency to the entire belief set.

1.1.2 Contraction

A contraction occurs when the agent stops believing in certain information. In simple terms, a *contraction* consists of giving up as many beliefs as needed so that the new belief set no longer implies the unwanted belief. Unlike expansion, contractions are characterized by postulates that capture the underlying intuition behind the operation (Alchourrón, Gärdenfors, *et al.*, 1985). Let's define a contraction $K - \varphi$, where K is the initial belief set, and φ is the belief to be removed.

(Closure) The result of contracting a belief set is always another belief set:

$$K - \varphi \textit{ is a belief set}$$

(Inclusion) Contractions cannot add new beliefs to the belief set during the operation:

$$K - \varphi \subseteq K$$

(Vacuity) If the unwanted belief is not present in the belief set, no changes occur to the set:

$$\textit{if } \varphi \notin K, \ \textit{then } K - \varphi = K$$

(Success) On the other hand, unless a belief is a tautology (hence, an element of every belief set), a contraction operation will always remove the unwanted belief:

$$\textit{if } \nvdash \varphi, \ \textit{then } \varphi \notin (K - \varphi)$$

(Recovery) A contraction is recoverable through an expansion operation:

$$K \subseteq (K - \varphi) + \varphi$$

(Equivalence) Contractions made by equivalent beliefs yield equivalent results:

$$\textit{if } \vdash \varphi \leftrightarrow \psi, \ \textit{then } (K - \varphi) = (K - \psi)$$

Furthermore, besides the basic postulates, the AGM model defines two additional ones:

(Conjunctive overlap) A belief ω that is part of $(K - \varphi)$ and $(K - \psi)$ will also be part of $K - (\varphi \wedge \psi)$:

$$(K - \varphi) \cap (K - \psi) \subseteq K - (\varphi \wedge \psi)$$

(Conjunctive inclusion) Any belief deleted when removing φ will also be deleted when removing $\varphi \wedge \psi$:

$$\textit{if } \varphi \notin K - (\varphi \wedge \psi), \ \textit{then } K - (\varphi \wedge \psi) \subseteq (K - \varphi)$$

1.1.3 Revision

The revision process involves adding new information to the belief set while ensuring that the final belief set will be consistent. If a belief set remains consistent after adding a new belief, the revision operation will add the belief to the set without modification. However, if the expansion results in inconsistency, other parts of the initial belief set may need to be removed to achieve consistency. Like contraction, the revision operation $K * \varphi$ is characterized by a set of basic postulates.

(Closure) Just like in contraction, the outcome of a revision in a belief set always results in another belief set:

$$K * \varphi \textit{ is a belief set}$$

(Success) The success of a revision is indicated by a resulting belief set that contains the revised belief:

$$\varphi \in (K * \varphi)$$

(Inclusion) No information besides the revised belief and its consequences will be added to the belief set during an operation of revision:

$$(K * \varphi) \subseteq (K + \varphi)$$

(Preservation) All the beliefs are preserved if the new belief is consistent with the belief set. In this case, expansion and revision are equivalents:

$$\textit{if } \neg\varphi \notin K, \ \textit{then } (K + \varphi) \subseteq (K * \varphi)$$

(Consistency) The rational agent should aim for consistency at any cost. So, unless the new information in itself is inconsistent, the revised belief set is consistent:

$$\textit{if } \varphi \textit{ is consistent, then } (K * \varphi) \textit{ is also consistent}$$

(Equivalence) Revisions made by equivalent beliefs yield equivalent results:

$$\textit{if } \vdash \varphi \leftrightarrow \psi, \ \textit{then } (K * \varphi) = (K * \psi)$$

Finally, revision also has two extra postulates to deal with revisions by conjunctions. They say that for any two sentences φ and ψ, if in revising the initial belief set K by φ, one is lucky enough to reach a belief set $K * \varphi$ consistent with ψ, then to produce $K * (\varphi \wedge \psi)$, all that one needs to do is to expand $K * \varphi$ with ψ (Peppas, 2008).

(Super-expansion) $K * \varphi$ is a minimal change of K to include φ, and therefore, there is no way to arrive at $K * (\varphi \wedge \psi)$ from K with "less change":

$$K * (\varphi \wedge \psi) \subseteq (K * \varphi) + \psi$$

(Sub-expansion) However, if ψ is consistent with $(K * \varphi)$, these further changes can be limited to simply adding ψ to $(K * \varphi)$ and closing under logical implications:

$$\textit{if}\, \neg\psi \notin (K * \varphi) \;\textit{then}\; (K * \varphi) + \psi \subseteq K * (\varphi \wedge \psi)$$

1.2 Partial meet contraction

The postulates do not determine a unique contraction or revision operation for a belief set but rather characterize such operations. Alchourrón, Gärdenfors, *et al.* (1985) introduce a construction for contraction functions based on the idea of selecting maximal subsets that do not entail the belief being contracted. This operation is known as partial meet contraction.

In *partial meet contraction*, a selection function selects maximal subsets from the belief set that do not entail the sentence being contracted. In other words, partial meet contraction is defined as the intersection of all sets selected by the selection function. This construction aims to minimize the loss of information from the belief set during contraction. A partial meet contraction satisfies the six basic postulates of contraction in the AGM model.

Definition 1.1. Let K be a belief set and φ be a proposition. Alchourrón and Makinson (1981) define a *remainder set*, denoted by $\mathsf{rem}(K, \varphi)$, as the family of all maximal subsets $B \subseteq K$ that do not imply φ. In other words, whenever $B \subseteq K$, B is a *remainder* of K with respect to φ if and only if both $\varphi \notin \mathsf{Cn}(B)$, and whenever $B' \subseteq K$ and $\mathsf{Cn}(B') \cap \varphi = \emptyset$ then $B \not\subset B'$.

Definition 1.2. Let K be a belief set and let γ be any function such that for every proposition φ, $\gamma(\mathsf{rem}(K, \varphi))$ is a non-empty subset of $\mathsf{rem}(K, \varphi)$, if the remainder set is non-empty, and $\gamma(\mathsf{rem}(K, \varphi)) = \{K\}$ in the limiting case that the remainder set is empty (Alchourrón, Gärdenfors, *et al.*, 1985). Such a function is a *selection function* for K.

Definition 1.3. Given the set of propositions K and the selection function γ, the operation of *partial meet contraction* of the proposition φ is defined by putting

$$K -_{\gamma} \varphi = \bigcap \gamma(\mathsf{rem}(K, \varphi))$$

for all φ (Alchourrón, Gärdenfors, *et al.*, 1985).

Definition 1.4. The limiting case when for all propositions φ, the selection function γ always picks up only one element of its argument is called *maxichoice contraction operation* (Alchourrón, Gärdenfors, *et al.*, 1985).

1.3 Belief bases

The AGM paradigm treated all beliefs in a belief set equally, making no distinction between basic beliefs and those inferred from the basic ones. Hansson (1991) proposed a significant generalization by introducing belief bases. A *belief base* is a set of propositions that is not closed under logical consequence. Its elements represent beliefs held independently of any other belief or set of beliefs (Hansson, 1991). This way, models utilizing belief bases enhance expressiveness compared to the belief set presented in Alchourrón, Gärdenfors, and Makinson (1985).

Hansson further generalized partial meet contraction to apply to belief bases. It is important to note that while partial meet contraction relies on selecting maximal subsets that do not imply the contracted belief, it is not sufficient for these subsets that they do not contain the contracted sentence. Hansson characterizes partial meet base contraction in terms of the following postulates:

(Success) Unless the belief is a tautology, the contracted belief will not be part of the set of consequences derived from the contracted belief base:

$$\textit{if } \nvdash \varphi, \textit{ then } \varphi \notin \mathsf{Cn}(B - \varphi)$$

(Inclusion) New beliefs will never be added to the belief base during the contraction:

$$B - \varphi \subseteq B$$

(Relevance) Blocks the exclusion of elements that there is no good reason to exclude:

$$\begin{gathered} \textit{if } \psi \in B \textit{ and } \psi \notin (B - \varphi), \textit{ then there is some } B' \textit{ such that} \\ (B - \varphi) \subseteq B' \subseteq B, \varphi \notin \mathsf{Cn}(B') \textit{ but } \varphi \in \mathsf{Cn}(B' \cup \{\psi\}) \end{gathered}$$

(Uniformity) Ensures that the result of contracting a sentence from a belief base only depends on which subsets of the belief base imply the contracted sentence. In other words, if all the subsets of B that imply some sentence φ also imply ψ, and vice versa, then the result of the contractions are equivalents:

$$\textit{if for all subsets } B' \textit{ of } B, \varphi \in \mathsf{Cn}(B') \textit{ if and only if } \psi \in \mathsf{Cn}(B'), \textit{ then } B - \varphi = B - \psi$$

Theorem 1.1. An operation of contraction of B is a *partial meet base contraction* if and only if it satisfies the four basic postulates above.

1.3.1 Kernel contraction

In Hansson (1994), Hansson introduced another construction for obtaining contraction operations, based on the notions of kernels and incision functions. Unlike partial meet contractions that aim for maximal subsets that do not imply the undesired belief, a kernel contraction targets the minimal belief sets that do imply the unwanted sentence.

As explained in (Wassermann, 2000), the idea behind the kernel contraction is that by removing at least one element from each minimal subset that implies the unwanted belief, we obtain a belief base that no longer implies this belief. To perform these removals, we use an incision function.

Definition 1.5. Let B be a belief base and let φ be a proposition. The *kernel set*, denoted by $\mathsf{ker}(B, \varphi)$, is the family of all minimal subsets $X \subseteq B$ that entail φ. In other words, whenever $X \subseteq B$, X is a *kernel* of B with respect to φ if and only if both $\varphi \in \mathsf{Cn}(X)$ and there is no $Y \subset X$ such that $\varphi \in \mathsf{Cn}(Y)$.

Definition 1.6. Let B be a belief base and let σ be any function such that for every proposition φ, it satisfies the following conditions:

- $\sigma(\mathsf{ker}(B, \varphi)) \subseteq \bigcup \mathsf{ker}(B, \varphi)$
- *if* X *is a non-empty element of* $\mathsf{ker}(B, \varphi)$, *then* $X \cap \mathsf{ker}(B, \varphi) \neq \emptyset$

Such a function is known as an *incision function*, which, in other words, is a function that selects at least one element from each kernel for removal.

Definition 1.7. Let σ be an incision function. The *kernel contraction* on B determined by σ is the operation $-_\sigma$ such that for all sentences φ:

$$B -_\sigma \varphi = B \setminus \sigma(\mathsf{ker}(B, \varphi))$$

In terms of postulates, instead of relevance, kernel contraction satisfies a slightly more general property called core-retainment:

> **(Core-retainment)** Requires of an excluded sentence ψ that it in some way contributes to the fact that B implies φ:
>
> $$\begin{aligned} &\textit{if } \psi \in B \textit{ and } \psi \notin (B - \varphi), \textit{ then there is some } B' \textit{ such that} \\ &\qquad B' \subseteq B \textit{ and } \varphi \notin \mathsf{Cn}(B') \textit{ but } \varphi \in \mathsf{Cn}(B' \cup \{\psi\}) \end{aligned}$$

Hansson also characterized kernel contraction in the terms of a set of postulates.

Theorem 1.2. An operation is a *kernel contraction* for a belief base if and only if it satisfies inclusion, success, uniformity and core-retainment (Hansson, 1994).

Broadly speaking, kernel contractions are more general than partial meet contractions. However, if the kernel contraction is induced by a minimal incision function, then it becomes a maxichoice contraction.

1.4 Pseudo-contraction

A downside of the contraction notions introduced in the AGM framework (Alchourrón, Gärdenfors, *et al.*, 1985), when applied to belief bases, is their syntax dependency, in the sense that they can only use sentences explicitly present in the belief base. For example, consider a belief base $B = \{p \wedge q\}$ where we want to get rid of the unwanted consequence q. The empty belief base would result from this contraction. However, by transforming B

into the equivalent belief base $B' = \{p, q\}$ would keep the belief p. This issue arises from the inclusion postulate, which forbids the addition of beliefs during a contraction.

In HANSSON (1989), a weakening of the inclusion postulate was introduced, termed *logical inclusion*.

(Logical inclusion) The consequences of a contraction are contained in the consequences of the belief base:

$$\mathsf{Cn}(B - \varphi) \subseteq \mathsf{Cn}(B)$$

Here, B represents a belief base, φ denotes the sentence to be contracted, and Cn refers to a consequence relation. By replacing the inclusion postulate, HANSSON allowed beliefs in the contraction's consequences to be added to the resulting belief base during the contraction.

Some years later, HANSSON proposed to call an operation that satisfies success and logical inclusion a *pseudo-contraction* (HANSSON, 1993a).

Definition 1.8. An operation is a *pseudo-contraction* if it satisfies the postulates of success and logical inclusion.

1.4.1 Cn* partial meet pseudo-contraction

SANTOS proposed a pseudo-contraction operation that depended on the formulas allowed to be added when contracting the belief base (SANTOS, 2016; SANTOS *et al.*, 2018). Following the results of HANSSON (1993a), SANTOS defined a construction of a pseudo-contraction operation called *Cn* partial meet pseudo-contraction* (SANTOS, 2016).

Definition 1.9. Let B be a belief base, Cn^* a consequence relation and γ a selection function. The *Cn* partial meet pseudo-contraction* is an operator such that for all φ:

$$B -^*_\gamma \varphi = \bigcap \gamma(\mathsf{rem}(\mathsf{Cn}^*(B), \varphi))$$

As detailed in SANTOS *et al.* (2018), Cn^* is a consequence relation that, in general, produces fewer consequences than the classical Cn. Let $B \subseteq \mathsf{Cn}^*(B) \subseteq \mathsf{Cn}(B)$, if $\mathsf{Cn}^*(B) = B$, a Cn^* pseudo-contraction becomes partial meet contraction for belief bases, and if $\mathsf{Cn}^*(B) = \mathsf{Cn}(B)$, it becomes partial meet contraction for belief sets. This operator generalizes those approaches, allowing us to get results between them if Cn^* varies in the range between B and $\mathsf{Cn}(B)$, which correspond to the thresholds for satisfaction of inclusion and logical inclusion, respectively.

As in the AGM paradigm, SANTOS characterizes a Cn^* partial meet pseudo-contraction by its postulates. He defined it as an operation that satisfies success and the following "starred" versions of inclusion, relevance, and uniformity:

(Inclusion*) Beliefs in the set of consequences of B concerning the consequence relation Cn^* can be part of the resulting contraction of the belief base:

$$B - \varphi \subseteq \mathsf{Cn}^*(B)$$

(Relevance*) As in the classical version, this postulate blocks the exclusion of elements that there is no good reason to exclude, but now considering the consequence relation Cn^*:

$$\begin{aligned}&\textit{if } \psi \in \mathsf{Cn}^*(B) \textit{ and } \psi \notin (B - \varphi), \textit{ then there is some } B' \textit{ such that}\\ &\qquad (B - \varphi) \subseteq B' \subseteq \mathsf{Cn}^*(B), \varphi \notin \mathsf{Cn}(B') \textit{ but } \varphi \in \mathsf{Cn}(B' \cup \{\psi\})\end{aligned}$$

(Uniformity*) Adapted from the classical uniformity postulate, if all subsets of $\mathsf{Cn}^*(B)$ that imply some sentence φ also imply ψ, and vice versa, then the result of the contractions are equivalent:

$$\textit{if for all subsets } B' \textit{ of } \mathsf{Cn}^*(B), \varphi \in \mathsf{Cn}(B') \textit{ if and only if } \psi \in \mathsf{Cn}(B'), \textit{ then } B - \varphi = B - \psi$$

As proved in Santos (2016), from the postulates follows the theorem:

Theorem 1.3. 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*.

1.4.2 Cn^* kernel pseudo-contraction

Analogously to Cn^* partial meet pseudo-contraction, Matos introduced Cn^* kernel pseudo-contraction by using kernels and incision functions for $\mathsf{Cn}^*(B)$ rather than B (Matos, 2021).

Definition 1.10. Let B be a belief base, Cn^* a consequence relation and σ an incision function. The *Cn^* kernel pseudo-contraction* of B by a sentence φ, is such that, for all sentences φ:

$$B -_\sigma^* \varphi = \mathsf{Cn}^*(B) \setminus \sigma(\mathsf{ker}(\mathsf{Cn}^*(B), \varphi))$$

To formalize the Cn^* kernel pseudo-contraction in terms of postulates, Matos first introduced the "starred" version of core-retainment:

(Core-retainment*) Similar to the classical version but considering the consequence relation Cn^*, this postulate requires of an excluded sentence ψ that it in some way contributes to the fact that B implies φ:

$$\begin{aligned}&\textit{if } \psi \in \mathsf{Cn}^*(B) \textit{ and } \psi \notin (B - \varphi), \textit{ then there is some } B' \textit{ such that}\\ &\qquad B' \subseteq \mathsf{Cn}^*(B) \textit{ and } \varphi \notin \mathsf{Cn}(B') \textit{ but } \varphi \in \mathsf{Cn}(B' \cup \{\psi\})\end{aligned}$$

As proved in Matos (2021), from the postulates the representation theorem follows:

Theorem 1.4. 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*.

1.4.3 Two-place pseudo-contraction

MATOS presented another kind of Cn^* pseudo-contraction using consequence operators that also considers the input formula.

Applying a two-place consequence operator to partial meet pseudo-contraction instead of using the classical one-place operator gives us the following definition:

Definition 1.11. Let B be a belief base, Cn^* a two place consequence operator and γ a selection function. The *Two-place Cn* partial meet pseudo-contraction* of B by a sentence φ, is such that, for all sentences φ:

$$B -^*_\gamma \varphi = \bigcap \gamma(\mathsf{rem}(\mathsf{Cn}^*(B, \varphi), \varphi))$$

In case the value returned by Cn^* does not depend on φ, then $\mathsf{Cn}^*(B) = \mathsf{Cn}^*(B, \varphi)$ and Cn^* is called a one-place consequence operator:

Proposition 1.1. Every Cn^* partial meet pseudo-contraction is a two-place Cn^* partial meet pseudo-contraction (SANTOS *et al.*, 2018).

As presented in SANTOS *et al.* (2018), the two-place Cn^* partial meet pseudo-contraction satisfies success, inclusion*, and relevance*, but uniformity* is lost.

Two-place Cn^* kernel pseudo-contraction are defined in MATOS (2021) analogously by using kernels and incision functions for $\mathsf{Cn}^*(B, \varphi)$ rather than for B:

Definition 1.12. Let B be a belief base, Cn^* a two place consequence operator and σ an incision function. The *two-place Cn* kernel pseudo-contraction* of B by a sentence φ is such that, for all sentences φ:

$$B -^*_\sigma \varphi = \mathsf{Cn}^*(B, \varphi) \setminus \sigma(\mathsf{ker}(\mathsf{Cn}^*(B, \varphi), \varphi))$$

As in Cn^* partial meet pseudo-contraction, this operation generalizes Cn^* kernel pseudo-contractions:

Proposition 1.2. Every Cn^* kernel pseudo-contraction is a two-place Cn^* kernel pseudo-contraction (SANTOS *et al.*, 2018).

Two-place Cn^* kernel pseudo-contraction satisfies success, inclusion*, and core-retainment*, but not uniformity*.

Chapter 2

Description Logics

Description logics (DLs) are a family of knowledge representation languages that can be used to represent knowledge of an application domain in a structured and well-understood way (Franz Baader, Horrocks, *et al.*, 2017). Most of them are decidable fragments of first-order logic. While first-order logic could be directly used to represent such knowledge, the fact that this logic is undecidable prevents the existence of an algorithm that decides whether the knowledge base entails a given sentence. A more or less expressive fragment influences how much knowledge can be expressed and the ability to perform reasoning tasks such as checking the satisfiability of concepts and the consistency of a knowledge base, subsumption, and answering different queries over the domain.

2.1 Foundations of Description Logics

The *signature* Σ of a description logic denotes the space available for defining concepts and assertions. It comprises the disjoint union of sets of *individual names* Σ_I, *concept names* Σ_C, and *role names* Σ_R:

$$\Sigma = \Sigma_I \,\dot{\cup}\, \Sigma_C \,\dot{\cup}\, \Sigma_R$$

In Description Logic, *concepts* are unary predicates of first-order logic built from concept names and role names using language-specific constructors. In contrast, *roles* are binary predicates built from role names that express the relation between individuals.

A *knowledge base (KB)* is a set of axioms categorized into terminological and assertional axioms.[1] *Terminological axioms (TBox)* restrict the interpretation of concepts and roles, representing the knowledge about the domain. *Assertional axioms (ABox)* limit individuals' interpretation, representing knowledge about a concrete situation. In other words, given a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$, a knowledge base $\mathcal{K}$ is:

$$\mathcal{K} = (\mathcal{T}, \mathcal{A})$$

[1] This work uses the terms *ontology* and *knowledge base* as synonyms.

A description logic is characterized in terms of interpretations. An *interpretation* $\mathcal{I}$ is a pair $\langle \Delta^{\mathcal{I}}, .^{\mathcal{I}} \rangle$ where $\Delta^{\mathcal{I}}$ is a non-empty set representing the domain and $.^{\mathcal{I}}$ is an interpretation function that maps every concept name $C \in \Sigma_C$ to a concept $C^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, every role name $r \in \Sigma_R$ to an element $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ of pairs of individuals and every individual $a \in \Sigma_I$ to an individual $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ (Franz Baader, Horrocks, *et al.*, 2017). This mapping is extended to concept descriptions according to the semantics of the constructors (Franz Baader, Horrocks, *et al.*, 2017).

An interpretation $\mathcal{I}$ is a *model of a TBox* $\mathcal{T}$ if it satisfies all its concept inclusions. In the same way, an interpretation $\mathcal{I}$ is a *model of an ABox* $\mathcal{A}$ if it satisfies its assertions. Finally, it is a *model of a knowledge base* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ if it is a model of $\mathcal{T}$ and $\mathcal{A}$.

A variety of axioms may be part of TBoxes, depending on the language used. For a comprehensive list of terminological and assertional axioms, please refer to Appendix A. Here, we will emphasize the general concept inclusion for the terminological part and concept and role assertions for the assertional part. The syntax and semantics of these axioms are defined as follows:

Name	**Syntax**	**Semantics**
Terminological axioms		
general concept inclusion (GCI)	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
Assertional axioms		
concept assertion	$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
role assertion	$r(a, b)$	$\langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in r^{\mathcal{I}}$

Table 2.1: *Terminological and assertional axioms (Franz Baader, Horrocks, et al., 2017).*

For instance, terminological axioms to express that a student is a person and that a teacher is a person who teaches in a course would appear as follows:

$$\begin{aligned} \mathsf{Student} &\sqsubseteq \mathsf{Person} \\ \mathsf{Teacher} &\sqsubseteq \mathsf{Person} \sqcap \exists \mathit{teaches}.\mathsf{Course} \end{aligned}$$

On the other hand, assertional axioms would be structured like this to express that CS is a course and Mark is a teacher who teaches in the CS course:

$$\begin{aligned} &\mathsf{Course}(\mathsf{CS}) \\ &\mathsf{Teacher}(\mathsf{Mark}) \\ &\mathit{teaches}(\mathsf{Mark}, \mathsf{CS}) \end{aligned}$$

2.1.1 Quantified ABoxes

Quantified ABoxes were first introduced in Franz Baader, Kriegel, *et al.* (2020) as an extension of usual ABoxes, replacing the set of individuals Σ_I with a set of objects Σ_O

that comprises not only individuals but also variables.

Formally, a *quantified ABox (qABox)* $\exists\mathcal{X}.\mathcal{A}$ is defined then as a finite set $\mathcal{X}$ of *variables* and the matrix $\mathcal{A}$, which is a finite set of concept assertions $C(u)$ and role assertions $r(u, v)$, where u and v can be either individuals or variables.

The set of individuals in a quantified ABox is denoted by the set of objects Σ_O after removing all variables:

$$\Sigma_I(\exists\mathcal{X}.\mathcal{A}) = \Sigma_O \setminus \mathcal{X}$$

Every ABox $\mathcal{A}$ is essentially a quantified ABox without variables.

In the same work, Franz Baader, Kriegel, *et al.* (2020) characterize a qABox in a first-order formula by taking the conjunction of the assertions in $\mathcal{A}$ and prefacing it with an existential quantifier prefix containing the variables in $\mathcal{X}$.

As a result, quantified ABoxes brought more expressivity to the knowledge bases, allowing the creation of assertional axioms that were not possible with conventional ABoxes. For example, building on the previous example, we could assert that Mark is a teacher who teaches in some course where the specific course is unknown or intentionally left unspecified:

$$\exists\{\,\mathsf{x}\,\}\cdot\{\,\mathsf{Teacher(Mark)},\ \textit{teaches}\mathsf{(Mark, x)}\,\}$$

Note that, in this instance, x represents a variable in the qABox, not an individual.

2.2 Description Logic families and extensions

There is a range of description logics and their extensions. They follow a naming convention that describes the operators allowed and their expressivity levels. These languages start from a basic logic and are incremented with extensions to enhance expressivity. Extensions introduce new constructors for complex concepts by adding new types of formulas.[2]

Among the base logics are the attributive language ($\mathcal{AL}$), the frame-based language ($\mathcal{FL}$), and the existential language ($\mathcal{EL}$). Table 2.2 provides a comparison of the concepts available in these languages.

[2] Refer to Appendix A for a detailed list of the main concepts and extensions.

Name	Syntax	$\mathcal{AL}$	$\mathcal{FL}$	$\mathcal{EL}$
atomic concept	A	✓	✓	✓
top	$\top$	✓	✓	✓
bottom	$\bot$	✓	✓	
conjunction	$C \sqcap D$	✓	✓	✓
atomic negation	$\neg A$	✓		
value restriction	$\forall r.C$	✓	✓	
limited existential qualification	$\exists r.\top$	✓	✓	✓
role restriction	$r_{\restriction C}$		✓	
existential restriction	$\exists r.C$			✓

Table 2.2: *Comparison of basic description logics $\mathcal{AL}$, $\mathcal{FL}$ and $\mathcal{EL}$, where A is a concept name, C and D are either a concept name or a complex concept and r stand for role names. Adapted from: Franz Baader, Horrocks, et al. (2017).*

The upcoming section will explore the $\mathcal{EL}$ description logic.

2.3 The Description Logic $\mathcal{EL}$

Like other DLs, a knowledge base in $\mathcal{EL}$ is structured using concepts, roles and individuals, with roles defining relationships among individuals. Using the $\mathcal{EL}$ constructors presented in Table 2.2, Franz Baader, Koopmann, and Kriegel (2023) describe the axioms of $\mathcal{EL}$ as follows.

An $\mathcal{EL}$ *atom* is either a concept name $A \in \Sigma_C$ or an existential restriction $\exists r.C$, where r is a role name and C is an $\mathcal{EL}$ concept description.

An $\mathcal{EL}$ *concept description* is defined inductively according to this grammar rule:

$$C ::= \top \mid A \mid C \sqcap C \mid \exists r.C$$

where A ranges over concept names and r ranges over role names.

An $\mathcal{EL}$ *TBox* is a finite set of $\mathcal{EL}$ *concept inclusions* $C \sqsubseteq D$ where C and D are $\mathcal{EL}$ concept descriptions. It represents subconcept-superconcept relationships between concept descriptions.

Finally, an $\mathcal{EL}$ *ABox* consists of a finite set of $\mathcal{EL}$ *concept assertions*. These assertions can be concept assertions $C(a)$ that relate individuals to concepts or role assertions $r(a, b)$ that relate individuals to others. In this context, a and b represent individuals, r stands for a role name, and C is a concept description. $\mathcal{EL}$ ABoxes can also be extended into $\mathcal{EL}$ *quantified ABoxes* by expanding the set of individuals Σ_I to a set of objects Σ_O that includes both individuals and variables.

From the definitions above, an $\mathcal{EL}$ *ontology* is a pair consisting of an $\mathcal{EL}$ *ABox* and an $\mathcal{EL}$ *TBox*.

2.3.1 Reasoning tasks in $\mathcal{EL}$

Reasoning tasks are interesting depending on the constructors available in the language. For instance, satisfiability is not an interesting problem in $\mathcal{EL}$ as the constructors that could cause unsatisfiability are absent. Similarly, subsumption cannot be reduced to unsatisfiability (Franz Baader, Horrocks, *et al.*, 2017).

The entailment task is an important technique in $\mathcal{EL}$, allowing us to determine if an object is an instance of a given concept. We say that a concept assertion $C(a)$ is *entailed* by an ABox $\mathcal{A}$ with respect to a TBox $\mathcal{T}$ if $C(a)$ is satisfied in all models of $\mathcal{A}$ and $\mathcal{T}$. This is denoted by $\mathcal{A} \models^{\mathcal{T}} C(a)$.

In the same way, instance checking is a commonly used task performed by reasoners to validate if an individual belongs to a class. Formally, an individual a is said to be an *instance of* C with respect to the ABox $\mathcal{A}$ if $\mathcal{A} \models C(a)$.

For subsumption, if an $\mathcal{EL}$ concept inclusion $C \sqsubseteq D$ is satisfied in every model of $\mathcal{T}$, we say that C is *subsumed* by D with respect to $\mathcal{T}$. If $C \sqsubseteq D$ is satisfied when the TBox is empty, we denote it as $C \sqsubseteq^{\emptyset} D$.

Entailment relations

Instead of comparing ontologies based on their models, it is possible to compare them with respect to the answers to the queries they entail (Franz Baader, Koopmann, Kriegel, and Nuradiansyah, 2021; Franz Baader, 2023).

For a given query language QL, we say that a qABox $\exists X.\mathcal{A}$ *QL-entails* a qABox $\exists Y.\mathcal{B}$ ($\exists X.\mathcal{A} \models_{\mathrm{QL}} \exists Y.\mathcal{B}$) if the queries entailed by $\exists Y.\mathcal{B}$ is entailed by $\exists X.\mathcal{A}$. We say that they are *QL-equivalent*, if they entail each other ($\exists X.\mathcal{A} \equiv_{\mathrm{QL}} \exists Y.\mathcal{B}$) (Franz Baader, Koopmann, Kriegel, and Nuradiansyah, 2021).

For $\mathcal{EL}$ ontologies, Franz Baader, Koopmann, Kriegel, and Nuradiansyah (2021) consider two types of queries: instance queries and conjunctive queries:

- *Instance queries (IQ)* are $\mathcal{EL}$ concept descriptions viewed as first-order formulae with one free variable.
- *Conjunctive queries (CQ)* are quantified ABoxes with their individuals viewed as free variables. Due to this close connection, it is an easy consequence that the classical entailment relation between qABoxes coincides with CQ-entailment.

Chapter 3

Ontology Repair

In ontology engineering, we might need to repair a knowledge base either when modeling errors are detected or when it is necessary to remove an undesired formula (Franz Baader, 2023). The central question in ontology repair studies is then how to repair a knowledge base, removing the unwanted information while ensuring that:

(a) no new information is introduced;

(b) the ontology remains as close as possible to the original.

3.1 Classical approaches

Classical repair approaches define a repair as a maximal subset of the knowledge base that does not imply unwanted sentences (Franz Baader, Koopmann, Kriegel, and Nuradiansyah, 2021). These approaches ensure that no new axioms are added during the repair process, maintaining the ontology as close as possible to its original by adhering to the maximality condition (Franz Baader, 2023). Classical methods are based on axiom pinpointing (Schlobach and Cornet, 2003), which provides methods to understand why a particular consequence holds by computing its justifications.

Definition 3.1. A *justification* is a minimal subset of the knowledge base that entails the consequence in question.

From its justifications, classical repairs can be obtained by removing a minimal hitting set from the justifications of a sentence.

Definition 3.2. Given a collection $\{X_1, ..., X_k\}$ of sets, a *hitting set* $\mathcal{H}$ for this collection is a subset of $X_1 \cup ... \cup X_k$ such that $\mathcal{H} \cap X_i \neq \emptyset$ for all $i = 1, ..., k$. This hitting set is *minimal* if no other hitting set is strictly contained in it (Reiter, 1987).

In other words, repairs are computed, ensuring that no justification holds in the computed subsets of the knowledge base. Formally, a repair is defined as follows:

Definition 3.3. A *repair* of a knowledge base $\mathcal{K}$ with respect to the sentence α is a set $\mathcal{K}'$ such that $\mathsf{Cn}(\mathcal{K}') \subseteq \mathsf{Cn}(\mathcal{K})$ and $\mathcal{K}'$ does not entail α, i.e., $\alpha \notin \mathsf{Cn}(\mathcal{K}')$.

Definition 3.4. An *optimal repair* is a repair such that there is no other repair which contains it.

Classical repairs are defined as follows:

Definition 3.5. A *classical repair* of a knowledge base $\mathcal{K}$ with respect to α is repair $\mathcal{K}'$ such that $\mathcal{K}' \subset \mathcal{K}$

Definition 3.6. An *optimal classical repair* is a classical repair such that there is no other classical repair which contains it.

While these approaches preserve as many sentences as possible, the obtained results may not preserve a maximal amount of consequences and are syntax-dependent. For instance, suppose we have the sentence $(A \sqcap B \sqcap C)(a)$, and we want to remove $A(a)$ from its consequences. In this case, the repaired ontology would be the empty set, resulting in the loss of $(B \sqcap C)(a)$.

To overcome this problem, gentle repair approaches have been introduced. Instead of completely removing certain axioms, these approaches replace them with weaker ones. As defined in Franz Baader, Kriegel, *et al.* (2018):

Definition 3.7. An axiom α is *weaker* than β if $\mathsf{Cn}(\{\alpha\}) \subset \mathsf{Cn}(\{\beta\})$.

Gentle repairs assume that, when repairing, only the refutable part is susceptible to changes. Using Definition 3.7, Franz Baader, Kriegel, *et al.* (2018) define gentle repairs as:

Definition 3.8. Let $\mathcal{K} = \mathcal{K}_s \cup \mathcal{K}_r$ be a knowledge base and let α be a sentence entailed by $\mathcal{K}$ but not by $\mathcal{K}_s$. A knowledge base $\mathcal{K}' = \mathcal{K}_s \cup \mathcal{K}_r'$ is a *gentle repair* of $\mathcal{K}$ with respect to α if it is a repair of $\mathcal{K}$ and for each axiom $\varphi \in \mathcal{K}_r'$, φ is either an axiom from $\mathcal{K}_r$ or a weaker version of a removed axiom.

However, as shown in Franz Baader, Kriegel, *et al.* (2018), this approach still need not produce optimal repairs, i.e., ones that preserve a maximal set of consequences, and they are also dependent on the syntactic form of the axioms.

3.1.1 Classical repair approaches and pseudo-contractions

Although the studies of contractions in Belief Revision and the repairs in Ontology Engineering aim to solve the same issue, they approach the problem in different ways. Belief Revision focuses on the characteristics of what a contraction is, while Ontology Repair concentrates on finding ways to compute an optimal repair. For example, in Belief Revision, there is no restriction on a belief base being a finite set, whereas knowledge bases in ontology engineering are usually assumed to be finite.

In Santos *et al.* (2018) and Matos *et al.* (2019), the authors explored the similarities between contractions and repairs. Matos *et al.* (2019) demonstrated the equivalence between kernels and justifications. They also established a correspondence between classical repairs and remainders in partial meet base contractions and gentle repairs and pseudo-contractions.

Franz Baader (2023) also studied the similarities of these areas. As the notions of optimal classical repairs and remainders coincide, he showed that a maxichoice partial meet contraction operation is an optimal classical repair whenever the unwanted sentence is not a tautology and, if it is, the operation does not change the base.

3.2 Functional approaches for ABox repair

Functional approaches to produce ontology repairs were developed to create a method of repair where the syntactic structure of the axioms in the ontology was supposed to be irrelevant. These approaches shift the focus from the axioms to their consequences, interpreting the repair conditions as:

(a) no new consequences are introduced;

(b) a maximal set of consequences is preserved rather than a maximal set of axioms.

In Franz Baader, Koopmann, Kriegel, and Nuradiansyah (2021), the authors proposed methods to compute optimal repairs, considering the use of quantified ABoxes in scenarios where the initial qABox might contain errors while assuming the TBox to be static. They introduced different notions of repair, replacing the idea of containment of classical repairs with the notions of entailment.

In these terms, Franz Baader, Koopmann, Kriegel, and Nuradiansyah (2021) defined an *optimal repair* as an ontology that does not have unwanted consequences, is entailed by the original ontology, and preserves the maximal amount of implications in the sense that there is no repair that strictly entails it. In the same work, the authors also demonstrated that, in general, optimal repairs need not exist, even if there are repairs. Moreover, unlike classical repair approaches, even if optimal repairs exist, they need not cover all repairs in the sense that every repair is contained in an optimal repair.

The remainder of this section will explore the constructions of repairs introduced in a series of works (Franz Baader, Kriegel, *et al.*, 2020; Franz Baader, Koopmann, Kriegel, and Nuradiansyah, 2021; Franz Baader, Koopmann, Kriegel, and Nuradiansyah, 2022; Franz Baader, Koopmann, and Kriegel, 2023).

3.2.1 Canonical Repair

Canonical repair is a method where copies of individuals and their assertions are created while removing unwanted consequences. Each copy replicates the original individual, preserving information that would otherwise be lost when removing certain consequences. For example, given the assertions:

$$\mathsf{Scottish(Amy)}$$
$$\mathsf{Traveler(Amy)}$$

If we aim to remove the assertion that Amy is Scottish, we would also lose the information that there is a Scottish person who is a traveler. However, by creating a copy of Amy as a free variable x, we can retain this knowledge:

Traveler(Amy)
Scottish(x)
Traveler(x)

The unwanted sentence Scottish (Amy) was removed, but the consequence "There is a Scottish who is a traveler" was retained. Inevitably, creating copies extends the initial ABox to a quantified ABox, where the added variables serve as copies of the original objects.

Canonical repair is restricted to the case where the TBox is assumed to be correct and thus will not change during the process. However, the construction still considers the existence of terminological axioms to ensure that the removed consequences are not reintroduced by them (Franz Baader, Koopmann, Kriegel, and Nuradiansyah, 2021).

To explain the process of constructing a canonical repair, Franz Baader, Koopmann, Kriegel, and Nuradiansyah (2021) first define what is a repair request.
Definition 3.9. A *repair request* is the set of unwanted consequences defined as a finite set of $\mathcal{EL}$ concept assertions (Franz Baader, Koopmann, Kriegel, and Nuradiansyah, 2021).

The process for computing a canonical repair $\exists Y.\mathcal{B}$ of the ABox $\mathcal{A}$ with respect to the TBox $\mathcal{T}$ and the repair request $\mathcal{R}$, can be described in the following steps:

Saturation

The *saturation* process extends the given ABox with the consequences entailed by the TBox (Franz Baader, Koopmann, Kriegel, and Nuradiansyah, 2021). The rules applied during this step depend on the query language used during the repair.

For conjunctive queries, the *CQ-saturation rules* are defined as follows:

$\sqcap$-rule: If $(C_1 \sqcap \cdots \sqcap C_n)(t) \in \mathcal{A}$, then replace this assertion with $C_1(t), \cdots, C_n(t)$.

$\exists$-rule: If $(\exists r.C)(t) \in \mathcal{A}$, then replace this assertion with the pair of assertions $r(t,x)$ and $C(x)$, and add x to X, where x is a fresh variable not occurring in $\mathcal{A}$ or X.

$\sqsubseteq$-rule: If $C \sqsubseteq D \in \mathcal{T}$ and $\mathcal{A} \models C(t)$, and $\mathcal{A} \not\models D(t)$, then add $D(t)$ to $\mathcal{A}$.

While the third rule adds new concept assertions implied by the ABox and the TBox, the other two rules are responsible for breaking down complex concept assertions into atomic parts. The $\sqcap$-rule has the highest priority, followed by the $\exists$-rule and the $\sqsubseteq$-rule.

Franz Baader, Koopmann, Kriegel, and Nuradiansyah (2021) demonstrated that applying these rules may not always terminate, but for cycle-restricted TBoxes, CQ-saturation always terminates.

For instance queries, the *IQ-saturation rules* are slightly different:

$\sqcap$-rule: If $(C_1 \sqcap \cdots \sqcap C_n)(t) \in \mathcal{A}$, then replace this assertion with $C_1(t), \cdots, C_n(t)$.

$\exists$-rule: If $(\exists r.C)(t) \in \mathcal{A}$, then replace this assertion with the pair of assertions $r(t, x_C)$ and $C(x_C)$, and add x_C to X if it is not already there.

$\sqsubseteq$-rule: If $C \sqsubseteq D \in \mathcal{T}$ and $\mathcal{A} \models C(t)$, and $\mathcal{A} \not\models D(t)$, then add $D(t)$ to $\mathcal{A}$.

As shown in Franz Baader, Koopmann, Kriegel, and Nuradiansyah (2021), the IQ-saturation always terminates without imposing any restriction on the TBox. The key difference lies in the fact that IQ-saturation rules are more parsimonious when introducing new objects, ensuring that added variables are linked to existing concepts. For instance, when applying the $\exists$-rule to an assertion $\exists r.C(t)$, the IQ-saturation adds a new fresh variable x_C only if it is not contained in $\mathcal{X}$; otherwise, it reuses the existing in the new assertion. This restriction ensures that the saturation will not be trapped in a cycle, thereby ensuring it will terminate.

Regardless of the chosen query language, *saturation* is the process of exhaustively applying the respective rules to break down complex concepts and add consequences of the TBox into the ABox. This process results in a quantified ABox formed by atomic concepts, making it suitable for repair since the TBox information is included within the resulting qABox.

Copying

The process of *copying* generates copies of the objects in the saturated ABox. Each object u in the input qABox $\exists \mathcal{X}.\mathcal{A}$ receives copies of the form $\langle u, \mathcal{M} \rangle$ where $\mathcal{M}$ specifies which assertions $C(u)$ entailed by $\mathcal{A}$ must not hold for this copy (Franz Baader, Koopmann, and Kriegel, 2023). Formally, this set is defined as follows:

Definition 3.10. A *repair type* $\mathcal{M}$ for u is a subset of the set of atoms occurring in either the TBox $\mathcal{T}$ or in the repair request $\mathcal{R}$ that satisfies the following properties:

- $\mathcal{A} \models C(u)$ for each atom $C \in \mathcal{M}$.
- $C \not\sqsubseteq^{\emptyset} D$ for each pair of distinct atoms C, D in $\mathcal{M}$.
- If C is an atom in $\mathcal{M}$ and $E \sqsubseteq F$ is an axiom in $\mathcal{T}$ with $\mathcal{A} \models E(u)$ and $F \sqsubseteq^{\emptyset} C$, then there is an atom D in $\mathcal{M}$ such that $E \sqsubseteq^{\emptyset} D$.

The first condition ensures that u is an instance of the atoms in $\mathcal{M}$. The second condition avoids duplicates, while the third ensures that the TBox will not reintroduce consequences removed due to $\mathcal{M}$.

Repairing

While the copying process generates multiple copies of each object, the *repairing* step will choose specific assertions for each new variable to preserve. These assertions will retain particular knowledge while avoiding unwanted consequences. Using the generated copies, Franz Baader, Koopmann, and Kriegel (2023) define the matrix of canonical repairs as follows:

Definition 3.11. The *matrix $\mathcal{B}$ of canonical repairs* consists of the following assertions:

- $A(\langle u, \mathcal{M} \rangle)$ if $A(u) \in \mathcal{A}$ and $A \notin \mathcal{M}$.

– $r(\langle u, \mathcal{M}\rangle, \langle v, \mathcal{N}\rangle)$ if $r(u,v) \in \mathcal{A}$ and for each $\exists r.C \in \mathcal{M}$ with $\mathcal{A} \models C(v)$, there is an atom $D \in \mathcal{N}$ such that $C \sqsubseteq^{\emptyset} D$.

These conditions ensure that the assertions added to $\mathcal{B}$ are not related to the atoms in their repair type. The intuition is that an object cannot be an instance of an unwanted consequence. Since the repair type is formed by atoms of unwanted consequences, by removing these assertions, the canonical repair ensures that the unwanted consequence no longer follows.

Selection

The result of the repairing step is a quantified ABox formed by copies of the original objects. The *selection* step chooses, for each individual a in the original qABox, one of its copies as representative of a in $\mathcal{B}$, in a way that the obtained qABox is really a repair (Franz Baader, Koopmann, and Kriegel, 2023). This choice is realized by fixing a repair seed:
Definition 3.12. A *repair seed* $\mathcal{S}$ maps each individual a to a repair type $\mathcal{S}_a$ for a such that the following condition is satisfied:

– If $C(a) \in \mathcal{R}$ and $\mathcal{A} \models C(a)$, then there is an atom D in $\mathcal{S}_a$ such that $C \sqsubseteq^{\emptyset} D$.

Given the repair seed $\mathcal{S}$, the *canonical repair* is the qABox $\exists Y.\mathcal{B}$ where the copies $\langle a, \mathcal{S}_a\rangle$ are replaced by their respective individual a. The other objects are variables in Y. This construction works both in the CQ and the IQ case, and yields a set of repairs that contains all repairs up to equivalence (Franz Baader, Koopmann, and Kriegel, 2023).

3.2.2 Canonical IQ-repair example

Consider the example where Amy loves an individual that is both Roman and a Nurse. This relationship could be represented by the following knowledge base:

(ABox) $\mathcal{A} = \{\ \exists \mathit{loves}(\mathsf{Roman} \sqcap \mathsf{Nurse})(\mathsf{Amy})\ \}$

(TBox) $\mathcal{T} = \emptyset$

Now, let's say that we want to get rid of this assertion. In this case, the repair request containing the unwanted consequence is defined as:

$$\mathcal{R} = \{\ \exists \mathit{loves}(\mathsf{Roman} \sqcap \mathsf{Nurse})(\mathsf{Amy})\ \}$$

The remainder of this section illustrates each step in the process of a canonical repair, as described in the previous section.

Saturation

Starting with the initial ABox $\mathcal{A}$, the saturation rules are applied as follows:

($\exists$-rule): $\exists X.\mathcal{A} = \exists\{\ x_{(\mathsf{R}\sqcap\mathsf{N})}\ \}\cdot\{\ \mathit{loves}(\mathsf{Amy}, x_{(\mathsf{R}\sqcap\mathsf{N})}), (\mathsf{Roman} \sqcap \mathsf{Nurse})(x_{(\mathsf{R}\sqcap\mathsf{N})})\ \}$

($\sqcap$-rule): $\exists X.\mathcal{A} = \exists\{\ x_{(\mathsf{R}\sqcap\mathsf{N})}\ \}\cdot\{\ \mathit{loves}(\mathsf{Amy}, x_{(\mathsf{R}\sqcap\mathsf{N})}), \mathsf{Roman}(x_{(\mathsf{R}\sqcap\mathsf{N})}), \mathsf{Nurse}(x_{(\mathsf{R}\sqcap\mathsf{N})})\ \}$

According to the $\exists$-rule from the IQ-saturation, the created variable should be named $x_{(\mathsf{R}\sqcap\mathsf{N})}$. However, since this is the only variable resulting from the saturation, we will name it x. The resulting qABox is then:

$$\exists \mathcal{X}.\mathcal{A} = \exists\{\ x\ \} \cdot \{\ \mathit{loves}(\mathsf{Amy}, x), \mathsf{Roman}(x), \mathsf{Nurse}(x)\ \}$$

Copying

The copying process creates copies of the objects in $\Sigma_O(\exists \mathcal{X}.\mathcal{A})$. Each copy is associated with a repair type (Definition 3.10) that specifies which atoms should be removed for this copy. In the current example, the atoms present in the sentences of the repair request $\mathcal{R}$ and the TBox $\mathcal{T}$ are:

$$\mathsf{Atoms}(\mathcal{R}, \mathcal{T}) = \{\ \mathsf{Roman}, \mathsf{Nurse}, \exists \mathit{loves}(\mathsf{Roman} \sqcap \mathsf{Nurse})\ \}$$

The possible repair types are given by the power set of the existing atoms:

$$\begin{aligned}
m_0 &:= \{\ \} & m_4 &:= \{\ \mathsf{Roman}, \mathsf{Nurse}\ \} \\
m_1 &:= \{\ \mathsf{Roman}\ \} & m_5 &:= \{\ \mathsf{Roman}, \exists \mathit{loves}(\mathsf{Roman} \sqcap \mathsf{Nurse})\ \} \\
m_2 &:= \{\ \mathsf{Nurse}\ \} & m_6 &:= \{\ \mathsf{Nurse}, \exists \mathit{loves}(\mathsf{Roman} \sqcap \mathsf{Nurse})\ \} \\
m_3 &:= \{\ \exists \mathit{loves}(\mathsf{Roman} \sqcap \mathsf{Nurse})\ \} & m_7 &:= \{\ \mathsf{Roman}, \mathsf{Nurse}, \exists \mathit{loves}(\mathsf{Roman} \sqcap \mathsf{Nurse})\ \}
\end{aligned}$$

For each object in $\Sigma_O(\exists \mathcal{X}.\mathcal{A})$, we use the definition of repair type (Definition 3.10) to check whether a repair type is valid for the given object and then create the new variables:

$$\begin{aligned}
y_0 &:= \langle \mathsf{Amy}, m_0 \rangle & y_2 &:= \langle x, m_0 \rangle \\
& \cancel{\langle \mathsf{Amy}, m_1 \rangle} & y_3 &:= \langle x, m_1 \rangle \\
& \cancel{\langle \mathsf{Amy}, m_2 \rangle} & y_4 &:= \langle x, m_2 \rangle \\
y_1 &:= \langle \mathsf{Amy}, m_3 \rangle & & \cancel{\langle x, m_3 \rangle} \\
& \cancel{\langle \mathsf{Amy}, m_4 \rangle} & y_5 &:= \langle x, m_4 \rangle \\
& \cancel{\langle \mathsf{Amy}, m_5 \rangle} & & \cancel{\langle x, m_5 \rangle} \\
& \cancel{\langle \mathsf{Amy}, m_6 \rangle} & & \cancel{\langle x, m_6 \rangle} \\
& \cancel{\langle \mathsf{Amy}, m_7 \rangle} & & \cancel{\langle x, m_7 \rangle}
\end{aligned}$$

By following these steps, we have computed y_0 and y_1 as copies of Amy, and y_2, y_3, y_4 and y_5 as copies of x.

Reparing

The repairing process will create the assertions for each copy based on the original object and the associated repair type. The generated assertions take into account the conditions described in the Definition 3.11:

$$\begin{array}{lll} \mathsf{Roman}(y_2) & \mathit{loves}(y_0, y_2) & \mathit{loves}(y_1, y_3) \\ \mathsf{Nurse}(y_2) & \mathit{loves}(y_0, y_3) & \mathit{loves}(y_1, y_4) \\ \mathsf{Nurse}(y_3) & \mathit{loves}(y_0, y_4) & \mathit{loves}(y_1, y_5) \\ \mathsf{Roman}(y_4) & \mathit{loves}(y_0, y_5) & \end{array}$$

Selection

Following the creation of new copies, the selection process will select, for each individual in $\Sigma_I(\exists \mathcal{X}.\mathcal{A})$, one of its copies as representative of this individual in the repaired qABox. In this example, there exists only one possible repair seed that follows the repair seed definition (Definition 3.12):

$$\mathcal{S}_{\mathsf{Amy}} := \{\, \exists \mathit{loves}(\mathsf{Roman} \sqcap \mathsf{Nurse}) \,\}$$

Given that m_3 is defined as $m_3 := \{\, \exists \mathit{loves}(\mathsf{Roman} \sqcap \mathsf{Nurse}) \,\}$, the copy of Amy selected to represent it in the repaired qABox corresponds to the one associated with m_3:

$$\mathsf{Amy} \iff y_1 := \langle \mathsf{Amy}, m_3 \rangle$$

The resulting repaired qABox is then:

$$\begin{aligned} \exists Y.\mathcal{B} = \exists\{\, y_0, y_2, y_3, y_4, y_5 \,\} \cdot \{\, & \mathsf{Roman}(y_2), \mathsf{Nurse}(y_2), \mathsf{Nurse}(y_3), \mathsf{Roman}(y_4), \\ & \mathit{loves}(y_0, y_2), \mathit{loves}(y_0, y_3), \mathit{loves}(y_0, y_4), \mathit{loves}(y_0, y_5), \\ & \mathit{loves}(\mathsf{Amy}, y_3), \mathit{loves}(\mathsf{Amy}, y_4), \mathit{loves}(\mathsf{Amy}, y_5) \,\} \end{aligned}$$

3.2.3 Optimized Repair

The canonical repair introduces more copies than necessary to achieve an optimal repair (Franz Baader, Koopmann, Kriegel, and Nuradiansyah, 2021). In contrast, the optimized repair approach presented in Franz Baader, Koopmann, Kriegel, and Nuradiansyah (2021) constructs an optimal repair using only the essential copies, ensuring that the resulting repair is equivalent to the canonical one.

All the processes of repair, such as *saturation*, *copying*, *repairing*, and *selection*, are identical to those described in the canonical approach. The key difference in the optimized approach is the addition of the *pruning* step. Instead of utilizing all the generated variables into the *repairing*, this step filters out redundant copies by selecting those associated with minimal cover repair types.

Before explaining the process of pruning, it is first necessary to define the concept of coverage:

Definition 3.13. Given two sets of concept descriptions $\mathcal{M}$ and $\mathcal{N}$, $\mathcal{N}$ *covers* $\mathcal{M}$ if each concept in $\mathcal{M}$ is subsumed by some concept in $\mathcal{N}$.

Pruning

Recalling Definition 3.11, an assertion $r(\langle u, \mathcal{M}\rangle, \langle v, \mathcal{N}\rangle)$ belongs to the resulting repair if the saturation contains $r(u, v)$ and for each $\exists r.C \in \mathcal{M}$ where $\mathcal{A} \models C(v)$, there is an atom $D \in \mathcal{N}$ such that C is subsumed by D.

In Franz Baader, Koopmann, Kriegel, and Nuradiansyah (2021), the authors rephrase it in terms of coverage: an assertion $r(\langle u, \mathcal{M}\rangle, \langle v, \mathcal{N}\rangle)$ belongs to the resulting repair if the saturation contains $r(u, v)$ and $\mathcal{N}$ covers $\mathrm{Succ}(\mathcal{M}, r, v)$, where Succ defines the set of concepts that v is an instance of, and is related to u through an existential restriction:

$$\mathrm{Succ}(\mathcal{M}, r, v) := \{\, C \mid \exists r.C \in \mathcal{M} \text{ and } \mathcal{A} \text{ entails } C(v) \,\}$$

The rules proposed in Franz Baader, Koopmann, Kriegel, and Nuradiansyah (2021) for constructing optimized repairs are dependent on the chosen query language. For the IQ-case, the construction starts with Y as an empty set of variables and proceeds by exhaustively applying the following rule:

(IQ-rule): The variable $\langle u, \mathcal{P}\rangle$ is added to Y if:

- $\langle t, \mathcal{M}\rangle$ is an element of $\Sigma_I \cup Y_i$
- The saturated ABox contains the role assertion $r(t, u)$
- $\mathcal{P}$ is a repair type for u
- $\mathcal{P}$ is a minimal cover of $\mathrm{Succ}(\mathcal{M}, r, u)$
- $\langle u, \mathcal{P}\rangle$ is not contained in $\Sigma_I \cup Y_i$

For the CQ-case, Y starts as the set of variables $\langle t, \emptyset\rangle$, where t is an object occurring in the saturated ABox and proceeds by exhaustively applying the following rule:

(CQ-rule): The variable $\langle u, \mathcal{P}\rangle$ is added to Y if:

- $\langle t, \mathcal{M}\rangle$ and $\langle u, \mathcal{N}\rangle$ are elements of $\Sigma_I \cup Y_i$
- The saturated ABox contains the role assertion $r(t, u)$
- The repair type $\mathcal{N}$ does not cover $\mathrm{Succ}(\mathcal{M}, r, u)$
- $\mathcal{P}$ is a repair type for u
- $\mathcal{P}$ is a minimal cover of $\mathcal{N} \cup \mathrm{Succ}(\mathcal{M}, r, u)$
- $\langle u, \mathcal{P}\rangle$ is not contained in $\Sigma_I \cup Y_i$

The intuition behind the pruning step is to select only the variables reachable from the initial individuals that have a repair type that minimally affects the knowledge base.

3.2.4 Optimized IQ-repair example

Reusing the same example from Section 3.2.2, this section illustrates each step of the optimized IQ-repair. The initial knowledge base remains the same:

(ABox) $\mathcal{A} = \{ \exists loves(\mathsf{Roman} \sqcap \mathsf{Nurse})(\mathsf{Amy}) \}$

(TBox) $\mathcal{T} = \emptyset$

With the following repair request:

$\mathcal{R} = \{ \exists loves(\mathsf{Roman} \sqcap \mathsf{Nurse})(\mathsf{Amy}) \}$

As previously discussed, the results from the *saturation* to the *selection* step are identical to those obtained in Section 3.2.2. From this point, we can proceed to the pruning process.

Pruning

The *copying* process created variables with valid repair types. In this step, unnecessary variables are pruned. For the IQ-repair, the construction begins with an empty set of variables and only one individual:

$Y_0 = \emptyset$

$\Sigma_I \cup Y_0 = \{ \mathsf{Amy} \}$

Recall that Amy is represented by the variable $\langle \mathsf{Amy}, m_3 \rangle$. The repeated application of the IQ-rules generates the following set of variables:

$Y_1 = Y_0 \cup \{ y_3 \}$, where $y_3 := \langle x, m_1 \rangle$

$Y_2 = Y_1 \cup \{ y_4 \}$, where $y_4 := \langle x, m_2 \rangle$

Since the rule cannot be reapplied, the final set of variables is used in the repair is:

$$Y := \{ y_3, y_4 \}$$

Repairing

The repairing step, then, will create the assertions for the remaining copies based on Definition 3.11. The process is the same as the canonical repair but now running over the results of the pruning step:

$$\begin{array}{ll} \mathsf{Nurse}(y_3) & loves(y_1, y_3) \\ \mathsf{Roman}(y_4) & loves(y_1, y_4) \end{array}$$

The resulting repaired qABox is then:

$$\exists Y.\mathcal{B} = \exists \{ y_3, y_4 \} \cdot \{ \mathsf{Nurse}(y_3), \mathsf{Roman}(y_4), loves(\mathsf{Amy}, y_3), loves(\mathsf{Amy}, y_4) \}$$

3.2.5 Explaining Optimal Repairs as Pseudo-contractions

In his work, Franz Baader (2023) explores the connection between optimal repairs and pseudo-contractions. By defining a consequence operator, he shows that, under certain conditions, optimal repairs can be obtained from pseudo-contractions.

Given the knowledge base $\mathcal{K}$ and the unwanted consequence φ, Franz Baader (2023) defines as a consequence operator Cn^*, the following two-place function:

$$\mathsf{Cn}^*(\mathcal{K}, \varphi) := \mathcal{K} \cup \bigcup \mathrm{O}_{\mathsf{rep}}(\mathcal{K}, \varphi)$$

where $\mathrm{O}_{\mathsf{rep}}(\mathcal{K}, \varphi)$ consists of the optimal repairs of $\mathcal{K}$ with respect to φ. This function is finite if and only if $\mathrm{O}_{\mathsf{rep}}(\mathcal{K}, \varphi)$ is finite for all knowledge bases $\mathcal{K}$ and sentences φ.

In the same work, it is proved that this function can effectively be computed since, as presented in Franz Baader, Koopmann, Kriegel, and Nuradiansyah (2022), optimal repairs can be computed by first computing the optimal quantified ABox repairs, which results in a finite and computable set.

Baader then proposes the following lemma as a connection between optimal repairs and remainders:

Lemma 3.1. Let $\mathcal{K}$ be a knowledge base and φ a sentence. If $\mathcal{B} \in \mathrm{O}_{\mathsf{rep}}(\mathcal{K}, \varphi)$, then $\mathcal{B}$ is equivalent to a remainder of $\mathsf{Cn}^*(\mathcal{K}, \varphi)$ with respect to φ.

As a consequence of this lemma, the following theorem is proved (Franz Baader, 2023):

Theorem 3.1. Let $\mathcal{K}$ be a knowledge base and φ a sentence. Then, there exists a Cn^* partial meet pseudo-contraction operation, such that:

$(\mathcal{K} -_\gamma \varphi)$ is an *optimal repair* if $\mathrm{O}_{\mathsf{rep}}(\mathcal{K}, \varphi) \neq \emptyset$

$(\mathcal{K} -_\gamma \varphi)$ is an *optimal classical repair* if $\mathrm{O}_{\mathsf{rep}}(\mathcal{K}, \varphi) = \emptyset$ and $\nvdash \varphi$

where the selection function γ is defined as a function that chooses an element of $\mathrm{O}_{\mathsf{rep}}(\mathcal{K}, \varphi)$ if this set is non-empty, and an arbitrary remainder of $\mathsf{Cn}^*(\mathcal{K}, \varphi)$ otherwise.

Remainders of $\mathsf{Cn}^*(\mathcal{K}, \varphi)$ are not necessarily optimal repairs, even if $\mathrm{O}_{\mathsf{rep}}(\mathcal{K}, \varphi) \neq \emptyset$. However, this is not a problem if $\mathrm{O}_{\mathsf{rep}}(\mathcal{K}, \varphi)$ covers[1] all repairs (Franz Baader, 2023). This coverage condition is satisfied if the ABox is restricted to being acyclic and the TBox to being cycle-restricted (Franz Baader, Koopmann, Kriegel, and Nuradiansyah, 2022; Franz Baader, 2023).

As a consequence, it was shown that a maxichoice Cn^* partial meet pseudo-contractions always produce optimal repairs in case $\mathrm{O}_{\mathsf{rep}}(\mathcal{K}, \varphi)$ covers all repairs (Franz Baader, 2023). Additionally, since every kernel contraction induced by a minimal incision function can be obtained as a maxichoice partial meet contraction, the results also hold for Cn^* kernel pseudo-contractions.

[1] Covers in the sense that there is an optimal repair that contains all repairs in it.

Chapter 4

Optimal Repairs and Pseudo-contractions

It is intuitive to see the similarities between contractions and ontology repair approaches. Previous studies, such as Matos *et al.* (2019) and Franz Baader (2023), have identified equivalences between pseudo-contractions operations and ontology repair methods. This chapter compares the results of contractions and repairs and proposes a method to minimize the differences between their results.

4.1 Comparing Optimal Repairs and Pseudo-contractions

Any of the approaches presented in Franz Baader, Koopmann, Kriegel, and Nuradiansyah (2021) to find an optimal ontology repair, whether through canonical or optimized repairs, use quantified ABoxes to increase the number of consequences preserved during the repair. However, as shown in Franz Baader, Koopmann, Kriegel, and Nuradiansyah (2022) and Franz Baader (2023), optimal ABox repairs are limited to the non-existence of cycles in the base. Consider, as an example, the following ontology and repair request:

Example 4.1. Given the ontology $\mathcal{O}$, defined by the TBox $\mathcal{T}$ and the ABox $\mathcal{A}$:

(TBox): $\mathcal{T} = \{ \exists r.B \sqsubseteq A, A \sqsubseteq \exists s.A, \exists s.A \sqsubseteq A \}$

(ABox): $\mathcal{A} = \{ r(a, b), B(b) \}$

(Repair Request): $\mathcal{R} = \{ A(a) \}$

In this example, the GCI $\exists s.A \sqsubseteq A$ together with $A \sqsubseteq \exists s.A$ characterizes a cycle, yielding a cyclic quantified ABox as a repair:

$$\exists Y.\mathcal{B} = \exists\{ x, y \} \cdot \{ r(a, x), B(b), s(a, y), s(y, y) \}$$

The resulting qABox has no equivalent optimal ABox repair as the cycle $s(y, y)$ prevents the removal of the variable y. While the quantified ABox enhanced the expressivity, it took the ontology out of the original language without giving the possibility of returning.

The main question is that knowledge bases represent a potentially infinite set of consequences. Once we look at this set and try to find a maximal subset that does not imply a particular formula, a finite representation of this maximal subset may not exist.

In contrast, the same knowledge base and repair request in a partial meet pseudo-contraction operation that uses the consequence operator of ELK[1] (Kazakov *et al.*, 2012), $\mathsf{Cn}_{\mathsf{ELK}}$, would result in the following set of remainders:

$$
\begin{aligned}
\mathsf{r}_1 &= \{\, A \sqsubseteq \exists s.A, \exists s.A \sqsubseteq A, \exists r.B \sqsubseteq A, r(a,b) \,\} \\
\mathsf{r}_2 &= \{\, A \sqsubseteq \exists s.A, \exists s.A \sqsubseteq A, \exists r.B \sqsubseteq A, B(b) \,\} \\
\mathsf{r}_3 &= \{\, A \sqsubseteq \exists s.A, \exists s.A \sqsubseteq A, B(b), r(a,b) \,\}
\end{aligned}
$$

This difference can be explained by the fact that, in Belief Revision, contractions consider the entire knowledge base as refutable, whereas constructing optimal repairs in the presented approaches assumes that terminological axioms cannot change.

In AGM theory (Alchourrón, Gärdenfors, *et al.*, 1985), beliefs are represented by belief sets, which are closed under logical consequence. Thus, a remainder is a maximal subset of the belief set that does not imply a specific formula, but, at the same time, because the sets are closed, it is also a maximal set with respect to preserving consequences, such as in optimal repairs.

In belief bases, remainders are maximal subsets of the base, i.e., of the finite representation, regardless of how they behave concerning the consequences. By using the classical consequence operator, a contraction could keep the maximum number of consequences since it includes all the existing consequences. However, it turns out that this is not a computable set. Reasoners only compute a closure of the consequences based on the tasks they can perform, such as classification, instance checking, or satisfiability checking. Otherwise, the number of inferences could become infinite, and the reasoner would never stop.

The limitation of reasoners in computing consequences also produces differences when comparing the results of pseudo-contractions and optimal repairs, even though both aim to find the maximal set of consequences. For instance, consider the example:

Example 4.2. Given the ABox $\mathcal{A}$ with an empty TBox $\mathcal{T}$, and the following repair request:

(TBox): $\mathcal{T} = \emptyset$

(ABox): $\mathcal{A} = \{\, r(a,b), A(b), B(b) \,\}$

(Repair Request): $\mathcal{R} = \{\, \exists r.(A \sqcap B)(a) \,\}$

The optimal repair computed by an optimized IQ-repair construction achieves one of the following quantified ABoxes:

---

[1] ELK is a specialized reasoner for the $\mathcal{EL}$ description logic. It can be found at: https://liveontologies.github.io/elk-reasoner/

$$\exists Y_1.\mathcal{B}_1 = \exists\{\ y_3\ \} \cdot \{\ r(a,b), A(b), r(a,y_3), B(y_3)\ \}$$
$$\exists Y_2.\mathcal{B}_2 = \exists\{\ y_4\ \} \cdot \{\ r(a,b), B(b), r(a,y_4), B(y_4)\ \}$$

These results are dependent on the chosen repair seed and can be expressed by equivalent ABoxes:

$$\mathcal{C}_1 = \{\ r(a,b), A(b), \exists r.B(a)\ \}$$
$$\mathcal{C}_2 = \{\ r(a,b), B(b), \exists r.A(a)\ \}$$

This example results in non-equivalent remainders when running the partial meet pseudo-contraction under consequences computed by the ELK reasoner:

$$\mathsf{r}_1 = \{\ r(a,b), A(b)\ \}$$
$$\mathsf{r}_2 = \{\ r(a,b), B(b)\ \}$$
$$\mathsf{r}_3 = \{\ A(b), B(b)\ \}$$

ELK did not compute any consequences for this base, which implies the lack of existential restrictions in its assertions, such as $\exists r.A(a)$ and $\exists r.B(a)$.

Optimal repairs offer the possibility of having more consequences in an extended version of the language while restricting the existence of cycles. On the other hand, pseudo-contractions are not concerned with cycles but depend on the closure of consequences computed by the reasoner. We fall between the absence of cycles or the dependency of reasoners. Let's explore one more example:

Example 4.3. Consider the ontology below along with the specified repair request:

(TBox): $\mathcal{T} = \{\ A \sqsubseteq B\ \}$

(ABox): $\mathcal{A} = \{\ r(a,b), A(b)\ \}$

(Repair Request): $\mathcal{R} = \{\ \exists r.(A \sqcap B)(a)\ \}$

This example illustrates the scenario where the TBox contributes to the justification of the unwanted sentence, and therefore, it needs to be considered during the repair process. The generation of variables provides a small set of copies for a and b:

$$\begin{array}{ll} a_0 := \langle a, \emptyset \rangle & b_0 := \langle b, \emptyset \rangle \\ a_1 := \langle a, \{\ \exists r.(A \sqcap B)\ \} \rangle & b_1 := \langle b, \{\ A\ \} \rangle \\ & b_2 := \langle b, \{\ A, B\ \} \rangle \end{array}$$

Note that only a_1 is a valid repair seed $\mathcal{S}_a$ for a, while any of the copies of b can be used as a repair seed $\mathcal{S}_b$ for b. It follows that, the optimized IQ-repair construction produce the following repairs depending on the chosen repair seed:

$$\begin{aligned}
\mathrm{O}_1 &= \{\ A \sqsubseteq B, \exists r.B(a), A(b), B(b)\ \} \\
\mathrm{O}_2 &= \{\ A \sqsubseteq B, r(a,b), B(b)\ \} \\
\mathrm{O}_3 &= \{\ A \sqsubseteq B, r(a,b), \exists r.B(a)\ \}
\end{aligned}$$

The first two repairs are optimal repairs for the given base and repair request. The third one is entailed by O_2 and, therefore, is not optimal.

Now, let's compare the optimized IQ-repair construction results with the remainders obtained from a partial meet pseudo-contraction operation using the same knowledge base and unwanted sentence. To do so, we will use the set of consequences computed by ELK:

$$\mathrm{Cn}_{\mathrm{ELK}}(\mathcal{O}) = \{\ A \sqsubseteq B, r(a,b), A(b), B(b)\ \}$$

From the inferred base, the remainders are as follows:

$$\begin{aligned}
\mathrm{r}_1 &= \{\ \mathrm{A} \sqsubseteq \mathrm{B}, \mathrm{A(b)}, \mathrm{B(b)}\ \} \\
\mathrm{r}_2 &= \{\ \mathrm{A} \sqsubseteq \mathrm{B}, r(\mathrm{a},\mathrm{b}), \mathrm{B(b)}\ \} \\
\mathrm{r}_3 &= \{\ r(\mathrm{a},\mathrm{b}), \mathrm{A(b)}\ \}
\end{aligned}$$

When comparing the remainders with the computed optimal repairs, they yield similar results. It is easy to see that r_2 is equal to O_2. Also, note that r_1 and O_1 are almost the same, differing only in the assertion $\exists r.B(a)$, which could not be computed due to the limitations of ELK in computing the set of consequences. Finally, the last remainder represents a contraction obtained by removing the axiom $A \sqsubseteq B$ from the TBox, which has no equivalent in optimal repairs, as the ontology repair constructions do not change the TBox.

The following section proposes an algorithm to extend the closure of consequences computed by the reasoner with a set of axioms entailed by the base.

4.2 An ontology saturator

Reasoners can only infer consequences based on their reasoning methods. To extend this set of consequences, we propose a new method to compute a specific family of consequences. Consider the following assertions:

$$\mathit{loves}(\mathrm{Amy}, \mathrm{Rory})$$

$$\mathrm{Nurse}(\mathrm{Rory})$$

The first assertion says that Amy loves Rory, while the second says that Rory is a nurse. Together, they imply that Amy loves a nurse. The following axiom can represent this:

$$\exists \mathit{loves}.\mathrm{Nurse}(\mathrm{Amy})$$

In other words, there exists a relationship *loves* between Amy and an individual who is an instance of the concept Nurse. We can also use the same construction to entail a chain of individual relations. For example, by adding the axioms:

$$\textit{friend}(\text{Doctor}, \text{Amy})$$
$$\text{Travaler}(\text{Doctor})$$

we can entail an axiom saying that Doctor is a friend of someone who loves a nurse, i.e.:

$$\exists \textit{friend}.\exists \textit{loves}.\text{Nurse}(\text{Doctor})$$

Axioms in a knowledge base can express the connections between individuals, and this form of axiom describes the chain of relations that connect a pair of individuals. If we imagine each individual in the base as a node and each axiom of the form $r(\text{a}, \text{b})$ as an edge $\langle \text{a}, \text{b} \rangle$, we can then create a graph of the relations existing in the base. In our example, we would have:

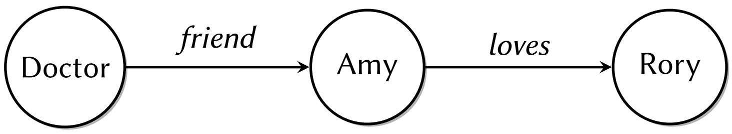

Figure 4.1: *Graph of the relations generated from the specified axioms*

By traversing the graph from a node to all reachable nodes, we can generate the existing chains of relations starting from the chosen node. Repeating this process for each node in the graph allows us to infer all the existing chains in the knowledge base. Using the depth-first search (DFS) algorithm, Program 4.1 and Program 4.2 describe the changes to compute the saturated knowledge base.

Program 4.1 The ontology saturator.

```
INPUT: Saturate the given ontology
OUTPUT: The saturated ontology
FUNCTION Saturate(𝒪)
    nodes ← ∅
    axioms ← ∅

    𝒢 ← CreateGraph(𝒪)
    for node ∈ 𝒢 do
        nodes[node] ← UNVISITED
    end

    for node ∈ 𝒢 do
        if nodes[node] = UNVISITED do
```

cont ⟶

```
⟶ cont
            axioms ← axioms ∪ DFS(node)
        end
    end
    return 𝒪 ∪ axioms
end
```

Program 4.1 starts with graph $\mathcal{G}$, constructed from the given ontology. The algorithm iterates through all the axioms in the ontology in order to build the graph. For each axiom of the form r(a, b), an edge ⟨a, b⟩ is added to the graph, where the edge stores the relationship r between the two individuals. Similarly, each node a includes a structure that stores all axioms of the form C(a) found during the iteration. The algorithm initializes all the nodes marked as unvisited, indicating that no node has been processed yet. Then, for each unvisited node in the graph, a DFS is performed starting from that node. The DFS function traverses the graph and collects new axioms based on the relationships encountered during the traversal. The original ontology is combined with the set of all encountered axioms, extending the ontology with the inferred axioms.

Program 4.2 The modified DFS.

```
INPUT: The graph of relations 𝒢 and the node where the search will start
OUTPUT: An array with the found axioms
FUNCTION DFS(𝒢, node)
    nodes[node] ← EXPLORED
    axioms ← ∅
    for u ∈ Adjacents(𝒢, node) do
        axioms ← axioms ∪ AddClassAxioms(node, u, roles)
        if nodes[u] = UNVISITED do
            axioms ← axioms ∪ AddChainAxioms(DFS(u), node, roles, u)
        end
    end
    return axioms
end
```

Program 4.2 presents the modified DFS algorithm. Iterating over all adjacent nodes of the current node, the algorithm runs a routine to generate the axioms related to these adjacent nodes (Program 4.3). Additionally, it also executes a routine to associate the generated chains of axioms from the recursive DFS calls with the current individual concerning the roles that connect them (Program 4.4).

Program 4.3 The class axioms generator.

```
INPUT: object, subject and the roles that connect them
OUTPUT: Axioms of the form ∃r.C found
FUNCTION AddClassAxioms(object, subject, roles)
    newAxioms ← ∅

    Classes ← GetClasses(object)
    for Class ∈ Classes do
        for r ∈ roles do
            newAxioms = newAxioms ∪ ∃r.Class(subject)
        end
    end

    return axioms ∪ newAxioms
end
```

Program 4.3 constructs new axioms based on the roles between the subject and the object individuals, also taking into account the concepts in which the object is an instance. Finally, Program 4.4 processes a set of axioms to generate new chain axioms based on the roles and the relationships between the individuals subject and object.

Program 4.4 The chain axioms generator.

```
INPUT: subject, object and the roles that connect them
OUTPUT: Axioms of the form ∃r.C found
FUNCTION AddChainAxioms(axioms, subject, object, roles)
    newAxioms ← ∅

    for a ∈ axioms do
        Class ← GetClass(axiom)

        if object is instanceOf(Class) do
            for r ∈ roles do
                newAxioms ← newAxioms ∪ ∃r.Class(subject)
            end
        end
    end

    return axioms ∪ newAxioms
end
```

It is important to note that infinite chains could occur depending on the existence of cycles in the graph. For instance, the addition of *friend*(Amy, Doctor) to the previous example would result in the following graph:

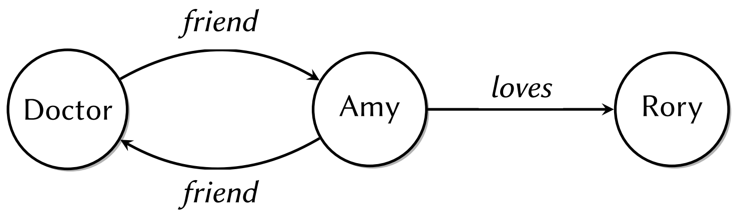

Figure 4.2: *Graph of relations generated after adding a cycle to the ontology*

This cycle generates the infinite chain $(\exists \mathit{friend}.)^n\top(\mathsf{Doctor})$. Note, however, that an infinite chain cannot be represented, as the algorithm would get stuck in infinite recursive calls. To avoid this, we use the cycle detection of DFS to represent only the first level of recursion within the cycle. Thus, the above cycle would generate the following chains:

$$\begin{array}{ll} \exists \mathit{friend}.\top(\mathsf{Doctor}) & \exists \mathit{friend}.\mathsf{Travaler}(\mathsf{Amy}) \\ \exists \mathit{friend}.\exists \mathit{friend}.\mathsf{Travaler}(\mathsf{Doctor}) & \exists \mathit{friend}.\exists \mathit{friend}.\top(\mathsf{Amy}) \end{array}$$

Going back to Example 4.3, consider using ELK to compute the consequences and then applying the saturation over the inferred ontology:

$$\mathsf{Cn}_{\mathsf{ELK}}(\mathcal{O}) = \{\ A \sqsubseteq B, r(a,b), A(b), B(b)\ \}$$
$$\mathsf{Sat}(\mathsf{Cn}_{\mathsf{ELK}}(\mathcal{O})) = \{\ A \sqsubseteq B, r(a,b), A(b), B(b), \exists r.A(a), \exists r.B(a)\ \}$$

The partial meet pseudo-contraction, now, results in the following remainders:

$$\begin{aligned} \mathrm{r}_1 &= \{\ \mathrm{A} \sqsubseteq \mathrm{B}, \exists r.\mathrm{B(a)}, \mathrm{A(b)}, \mathrm{B(b)}\ \} \\ \mathrm{r}_2 &= \{\ \mathrm{A} \sqsubseteq \mathrm{B}, r(\mathrm{a,b}), \mathrm{B(b)}, \exists r.\mathrm{B(a)}\ \} \\ \mathrm{r}_3 &= \{\ r(\mathrm{a,b}), \mathrm{A(b)}, \exists r.\mathrm{A(a)}, \exists r.\mathrm{B(a)}\ \} \\ \mathrm{r}_4 &= \{\ r(\mathrm{a,b}), \mathrm{B(b)}, \exists r.\mathrm{A(a)}, \exists r.\mathrm{B(a)}\ \} \\ \mathrm{r}_5 &= \{\ \mathrm{A(b)}, \mathrm{B(b)}, \exists r.\mathrm{A(a)}, \exists r.\mathrm{B(a)}\ \} \end{aligned}$$

Comparing the remainders to the results obtained by repairs, note that r_1 and r_2 are equivalent to the optimal repairs. All other remainders represent possible contractions obtained by changing the TBox. In this case, the existential restrictions added by the saturation process were sufficient to generate remainders equivalent to the optimal repairs. Let's examine another example:

Example 4.4. Given the following ontology and repair request:

(TBox): $\mathcal{T} = \{\ (A \sqcap C) \sqsubseteq B\ \}$

(ABox): $\mathcal{A} = \{\ r(a,b), A(b), C(b)\ \}$

(Repair Request): $\mathcal{R} = \{\ \exists r.B(a)\ \}$

Again, the TBox contributes to the justification of the unwanted sentence and therefore is considered to generate the copies:

$$
\begin{aligned}
a_0 &:= \langle a, \emptyset \rangle & b_0 &:= \langle b, \emptyset \rangle \\
a_1 &:= \langle a, \{ \exists r.B \} \rangle & b_1 &:= \langle b, \{ A \} \rangle \\
& & b_2 &:= \langle b, \{ C \} \rangle \\
& & b_3 &:= \langle b, \{ A, B \} \rangle \\
& & b_4 &:= \langle b, \{ A, C \} \rangle \\
& & b_5 &:= \langle b, \{ B, C \} \rangle \\
& & b_6 &:= \langle b, \{ A, B, C \} \rangle
\end{aligned}
$$

As the number of atoms increases, the number of variables generated and, consequently, the number of repairs will also increase. The only option of a repair seed for a is a_1. For the individual b, any generated variables can be used as a repair seed since no assertion of b is requested to be removed. It follows that the optimized IQ-repair is one of the following repairs:

$$
\begin{aligned}
\mathrm{O}_1 &= \{ (A \sqcap C) \sqsubseteq B, \exists r.A(a), \exists r.C(a), A(b), B(b), C(b) \} \\
\mathrm{O}_2 &= \{ (A \sqcap C) \sqsubseteq B, \exists r.A(a), \exists r.C(a), B(b), C(b) \} \\
\mathrm{O}_3 &= \{ (A \sqcap C) \sqsubseteq B, \exists r.A(a), \exists r.C(a), A(b), B(b) \} \\
\mathrm{O}_4 &= \{ (A \sqcap C) \sqsubseteq B, \exists r.A(a), \exists r.C(a), B(b) \} \\
\mathrm{O}_5 &= \{ (A \sqcap C) \sqsubseteq B, r(a, b), C(b), \exists r.A(a) \} \\
\mathrm{O}_6 &= \{ (A \sqcap C) \sqsubseteq B, r(a, b), A(b), \exists r.C(a) \} \\
\mathrm{O}_7 &= \{ (A \sqcap C) \sqsubseteq B, \exists r.A(a), \exists r.C(a) \}
\end{aligned}
$$

From the computed repairs, only O_1, O_5 and O_6 are optimal repairs. All the others are contained in O_1. Again, let's consider the computed remainders computed without the saturation process:

$$
\begin{aligned}
\mathrm{r}_1 &= \{ (\mathrm{A} \sqcap \mathrm{C}) \sqsubseteq \mathrm{B}, \mathrm{A}(\mathrm{b}), \mathrm{B}(\mathrm{b}), \mathrm{C}(\mathrm{b}) \} \\
\mathrm{r}_2 &= \{ (\mathrm{A} \sqcap \mathrm{C}) \sqsubseteq \mathrm{B}, r(\mathrm{a}, \mathrm{b}), \mathrm{C}(\mathrm{b}) \} \\
\mathrm{r}_3 &= \{ (\mathrm{A} \sqcap \mathrm{C}) \sqsubseteq \mathrm{B}, r(\mathrm{a}, \mathrm{b}), \mathrm{A}(\mathrm{b}) \} \\
\mathrm{r}_4 &= \{ r(\mathrm{a}, \mathrm{b}), \mathrm{A}(\mathrm{b}), \mathrm{C}(\mathrm{b}) \}
\end{aligned}
$$

In this example, no remainder is equivalent to an optimal repair. Although remainders and repairs are very similar, the remainders lack the consequences that the reasoner could not compute. By computing the consequences on ELK and then saturating the inferred ontology, the contraction operation will increase the number of consequences on the remainders:

$$
\mathrm{Cn}_{\mathrm{ELK}}(\mathcal{O}) = \{ (A \sqcap C) \sqsubseteq B, r(a, b), A(b), B(b), C(b) \}
$$

$$\mathsf{Sat}(\mathsf{Cn}_{\mathsf{ELK}}(\mathcal{O})) = \{\ (A \sqcap C) \sqsubseteq B, r(a,b), A(b), B(b), C(b) \exists r.A(a), \exists r.B(a), \exists r.C(a)\ \}$$

The remainders then, computed by a partial meet pseudo-contraction, are:

$$\begin{aligned} \mathrm{r}_1 &= \{\ (\mathrm{A} \sqcap \mathrm{C}) \sqsubseteq \mathrm{B}, \mathrm{A(b)}, \mathrm{B(b)}, \mathrm{C(b)}, \exists r.\mathrm{C}(a), \exists r.\mathrm{A}(a)\ \} \\ \mathrm{r}_2 &= \{\ (\mathrm{A} \sqcap \mathrm{C}) \sqsubseteq \mathrm{B}, r(\mathrm{a,b}), \mathrm{C(b)}, \exists r.\mathrm{C}(a), \exists r.\mathrm{A}(a)\ \} \\ \mathrm{r}_3 &= \{\ (\mathrm{A} \sqcap \mathrm{C}) \sqsubseteq \mathrm{B}, r(\mathrm{a,b}), \mathrm{A(b)}, \exists r.\mathrm{C}(a), \exists r.\mathrm{A}(a)\ \} \\ \mathrm{r}_4 &= \{\ r(\mathrm{a,b}), \mathrm{A(b)}, \mathrm{C(b)}, \exists r.\mathrm{C}(a), \exists r.\mathrm{A}(a)\ \} \end{aligned}$$

With the help of the reasoner, the saturator increased the closure of computed consequences so that the contraction achieved equivalence with the optimal repairs. The first three remainders are equivalent to the optimal repairs, while the last one has no equivalence, as it results from a contraction that changes the TBox.

The source code of the saturator developed for extending the ontology described in this section is available on GitHub.[2]

[2] https://github.com/davysouza/ontosat

Chapter 5

Conclusion

In this work, we studied the approaches to construct optimal repairs in Ontology Engineering and how they relate to the contractions in Belief Revision. Previous works researched the classical approaches of repairs, which presented methods to construct repairs by computing its justifications. Santos *et al.* (2018) and Matos *et al.* (2019) explored the similarities between classical repairs and pseudo-contractions. Starting from Franz Baader, Kriegel, *et al.* (2020), the authors introduced a series of works to propose functional approaches of repairs that were not dependent on the syntactic structure of the axioms. These approaches focused on repairing the ABoxes while assuming that TBoxes did not change. In Franz Baader (2023), the author compared optimal repairs resulting from functional constructions with pseudo-contractions, proving that in certain specific cases, optimal repairs were equivalent to the remainders of a partial meet pseudo-contraction. In this study, we have aimed to understand the scenarios in which the results of optimal repairs diverge from the remainders in contractions.

Compared to contractions, while the constructions of repairs presented in Franz Baader, Koopmann, Kriegel, and Nuradiansyah (2021) could achieve optimal results, the restriction to cycles and the fact that a repair does not affect the TBox can be seen as a pain point depending on the ontology that we need to repair. On the other hand, while functional approaches could find optimal repairs, the impossibility of a reasoner to compute an infinite set of consequences prevented the pseudo-contractions from achieving optimal results in some specific scenarios. To alleviate the differences between optimal repairs and pseudo-contraction operations, we proposed a method to saturate a knowledge base, expanding the closure of computed consequences. This approach proved to help achieve optimal repairs in bases where pseudo-contractions could not compute optimal results by using only the reasoner consequences.

It is unclear if the consequences added through the saturation are enough to ensure that pseudo-contractions consistently achieve optimal repairs. Also, it would be interesting to understand how functional approaches deal with description logics other than $\mathcal{EL}$ and if it is possible to expand functional approaches to cases where TBoxes can also change.

Appendix A

Description Logics

A.1 Description Logic constructors

Description logics are built from the constructors and axioms available in each language. Table A.1 presents a list of the most important concept and role constructors.

Name	Syntax	Semantics
Concept constructors		
atomic concept	A	$A^{\mathcal{I}}$
top	$\top$	$\Delta^{\mathcal{I}}$
bottom	$\bot$	$\emptyset$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
limited existential qualification	$\exists r.\top$	$\{ d \in \Delta^{\mathcal{I}} \mid \exists e \in \Delta^{\mathcal{I}} \text{ s.t. } \langle d, e \rangle \in r^{\mathcal{I}} \}$
existential restriction	$\exists r.C$	$\{ d \in \Delta^{\mathcal{I}} \mid \exists e \in C^{\mathcal{I}} \text{ s.t. } \langle d, e \rangle \in r^{\mathcal{I}} \}$
value restriction	$\forall r.C$	$\{ d \in \Delta^{\mathcal{I}} \mid \forall e \in \Delta^{\mathcal{I}}, \text{ if } \langle d, e \rangle \in r^{\mathcal{I}}, \text{ then } e \in C^{\mathcal{I}} \}$
unqualified number restriction	$(\leq nr)$	$\{ d \in \Delta^{\mathcal{I}} \mid \#\{e \in \Delta^{\mathcal{I}} \mid \langle d, e \rangle \in r^{\mathcal{I}} \} \leq n \}$
	$(\geq nr)$	$\{ d \in \Delta^{\mathcal{I}} \mid \#\{e \in \Delta^{\mathcal{I}} \mid \langle d, e \rangle \in r^{\mathcal{I}} \} \geq n \}$
qualified cardinality restriction	$(\leq nr.C)$	$\{ a \in \Delta^{\mathcal{I}} \mid \#\{ e \in C^{\mathcal{I}} \mid \langle d, e \rangle \in r^{\mathcal{I}} \} \leq n \}$
	$(\geq nr.C)$	$\{ a \in \Delta^{\mathcal{I}} \mid \#\{ e \in C^{\mathcal{I}} \mid \langle d, e \rangle \in r^{\mathcal{I}} \} \geq n \}$
nominal	$\{ a \}$	$\{ a^{\mathcal{I}} \}$
role value map	$(r \sqsubseteq s)$	$\{ d \in \Delta^{\mathcal{I}} \mid \text{ if } \langle d, e \rangle \in r^{\mathcal{I}} \text{ then } \langle d, e' \rangle \in s^{\mathcal{I}} \}$
Role constructors		
role composition	$r \circ s$	$\{ \langle d, f \rangle \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid \exists e \in \Delta^{\mathcal{I}}, \text{ if } \langle d, e \rangle \in r^{\mathcal{I}} \text{ and } \langle e, f \rangle \in s^{\mathcal{I}} \}$
inverse role	r^{-}	$\{ \langle d, e \rangle \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid \langle e, d \rangle \in r^{\mathcal{I}} \}$

Table A.1: *The key Description Logic concept and role constructors. Adapted from: Franz Baader, Horrocks, et al. (2017)*

A.2 Description Logic axioms

Knowledge bases consist of terminological and assertional axioms. Terminological axioms restrict the interpretation of concepts and roles, whereas assertional axioms restrict the interpretation of objects (F. Baader and Nutt, 2007). Table A.2 presents a list of the axioms that may be part of a knowledge base.

Name	Syntax	Semantics
Terminological concept axioms		
general concept inclusion (GCI)	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
concept definition	$C \equiv D$	$C^{\mathcal{I}} = D^{\mathcal{I}}$
Terminological role axioms		
role inclusion	$r \sqsubseteq s$	$r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$
role disjointness	$\mathsf{Disj}(r, s)$	$r^{\mathcal{I}} \cap s^{\mathcal{I}} = \emptyset$
role transitivity	$\mathsf{Trans}(r)$	$r^{\mathcal{I}}$ is transitive
role functionality	$\mathsf{Func}(r)$	$r^{\mathcal{I}}$ is functional
role reflexivity	$\mathsf{Ref}(r)$	$r^{\mathcal{I}}$ is reflexive
role irreflexivity	$\mathsf{Irref}(r)$	$r^{\mathcal{I}}$ is irreflexive
role symmetry	$\mathsf{Sym}(r)$	$r^{\mathcal{I}}$ is symmetrical
role antisymmetry	$\mathsf{Asym}(r)$	$r^{\mathcal{I}}$ is antisymmetrical
Assertional axioms		
concept assertion	$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
role assertion	$r(a, b)$	$\langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in r^{\mathcal{I}}$

Table A.2: *Terminological and assertional axioms. Adapted from: Franz Baader, Horrocks, et al. (2017)*

A.3 Description Logic extensions

In order to distinguish between different DLs, a naming scheme of some constructors and role axioms was introduced. These naming schemes starts from basic logics such as $\mathcal{AL}$ and $\mathcal{EL}$ and then each symbol added indicate a new constructor or axioms to the given basic DL.

The most common scheme starts with the basic DL $\mathcal{AL}$ (Franz Baader, Horrocks, *et al.*, 2017). For instance, build upon this, $\mathcal{ALC}$ is $\mathcal{AL}$ incorporated with negation. Note that $\mathcal{ALC}$ is equivalent to $\mathcal{ALCEU}$ since negation can be used to define disjunction and existential restriction.

Additionally, the symbol $\mathcal{S}$ is used as a shorthand for lengthy names. It represents $\mathcal{ALC}$ extended with transitivity.

Symbol	**Name**	**Syntax**
Concept constructors		
$\mathcal{U}$	disjunction	$C \sqcup D$
$\mathcal{C}$	negation	$\neg C$
$\mathcal{E}$	existential restriction	$\exists r.C$
$\mathcal{N}$	unqualified number restriction	$(\leq nr)$
		$(\geq nr)$
$\mathcal{Q}$	qualified number restriction	$(\leq nr.C)$
		$(\geq nr.C)$
$\mathcal{O}$	nominal	$\{ a \}$
Role constructors		
$\mathcal{I}$	inverse role	r^-
Role axioms		
$\mathcal{H}$	role inclusion	$r \sqsubseteq s$
$\mathcal{R}$	complex role inclusion	$r_1 \circ \cdots \circ r_n \sqsubseteq s$
$\mathcal{F}$	functionality	$\mathsf{Func}(r)$
$_{R+}$	transitivity	$\mathsf{Trans}(r)$

Table A.3: *DL extensions. Adapted from:* Franz Baader, Horrocks, et al. (2017)

References

[Alchourrón, Gärdenfors, *et al.* 1985] Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. “On the logic of theory change: partial meet contraction and revision functions”. *Journal of Symbolic Logic* 50.2 (1985), pp. 510–530. doi: 10.2307/2274239 (cit. on pp. 1, 3, 4, 6–8, 32).

[Alchourrón and Makinson 1981] Carlos E. Alchourrón and David Makinson. “Hierarchies of regulations and their logic”. In: *New Studies in Deontic Logic: Norms, Actions, and the Foundations of Ethics*. Ed. by Risto Hilpinen. Dordrecht: Springer Netherlands, 1981, pp. 125–148. isbn: 978-94-009-8484-4. doi: 10.1007/978-94-009-8484-4_5. url: https://doi.org/10.1007/978-94-009-8484-4_5 (cit. on p. 6).

[F. Baader and Nutt 2007] F. Baader and W. Nutt. “Basic description logics”. In: *The Description Logic Handbook: Theory, Implementation and Applications*. Ed. by Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F.Editors Patel-Schneider. Cambridge University Press, 2007, pp. 47–104 (cit. on p. 44).

[Franz Baader 2023] Franz Baader. “Optimal repairs in ontology engineering as pseudo-contractions in belief change”. In: *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*. SAC ’23. ACM, Mar. 2023. doi: 10.1145/3555776.3577719. url: http://dx.doi.org/10.1145/3555776.3577719 (cit. on pp. 1, 17, 19, 20, 28, 29, 31, 41).

[Franz Baader, Horrocks, *et al.* 2017] Franz Baader, Ian Horrocks, Carsten Lutz, and Uli Sattler. *An Introduction to Description Logic*. Cambridge University Press, Apr. 2017. isbn: 9780521695428. doi: 10.1017/9781139025355. url: http://dx.doi.org/10.1017/9781139025355 (cit. on pp. viii, ix, 1, 13, 14, 16, 17, 43–45).

[Franz Baader, Koopmann, and Kriegel 2023] Franz Baader, Patrick Koopmann, and Francesco Kriegel. “Optimal repairs in the description logic $\mathcal{EL}$ revisited”. In: *Logics in Artificial Intelligence*. Ed. by Sarah Gaggl, Maria Vanina Martinez, and Magdalena Ortiz. Cham: Springer Nature Switzerland, 2023, pp. 11–34. isbn: 978-3-031-43619-2 (cit. on pp. 16, 21, 23, 24).

[Franz Baader, Koopmann, Kriegel, and Nuradiansyah 2021] Franz Baader, Patrick Koopmann, Francesco Kriegel, and Adrian Nuradiansyah. “Computing Optimal Repairs of Quantified ABoxes w.r.t. Static $\mathcal{EL}$ TBoxes”. In: *Proceedings of the 28th International Conference on Automated Deduction (CADE-28), July 11–16, 2021, Virtual Event, United States*. Ed. by André Platzer and Geoff Sutcliffe. Vol. 12699. Lecture Notes in Computer Science. 2021, pp. 309–326. doi: https://doi.org/10.1007/978-3-030-79876-5_18 (cit. on pp. 1, 17, 19, 21–23, 26, 27, 31, 41).

[Franz Baader, Koopmann, Kriegel, and Nuradiansyah 2022] Franz Baader, Patrick Koopmann, Francesco Kriegel, and Adrian Nuradiansyah. “Optimal abox repair w.r.t. static $\mathcal{EL}$ tboxes: from quantified aboxes back to aboxes”. In: *The Semantic Web*. Ed. by Paul Groth *et al.* Cham: Springer International Publishing, 2022, pp. 130–146. isbn: 978-3-031-06981-9 (cit. on pp. 21, 29, 31).

[Franz Baader, Kriegel, *et al.* 2018] Franz Baader, Francesco Kriegel, Adrian Nuradiansyah, and Rafael Peñaloza. “Making repairs in description logics more gentle”. In: *Proceedings of the 31st International Workshop on Description Logics, Tempe, Arizona, October 27-29, 2018*. Ed. by Magdalena Ortiz and Thomas Schneider. Vol. 2211. CEUR-WS.org, Oct. 2018 (cit. on pp. 1, 20).

[Franz Baader, Kriegel, *et al.* 2020] Franz Baader, Francesco Kriegel, Adrian Nuradiansyah, and Rafael Peñaloza. “Computing compliant anonymisations of quantified aboxes w.r.t. $\mathcal{EL}$ policies”. In: *The Semantic Web – ISWC 2020*. Ed. by Jeff Z. Pan *et al.* Cham: Springer International Publishing, 2020, pp. 3–20. isbn: 978-3-030-62419-4 (cit. on pp. 1, 14, 15, 21, 41).

[Brachman and Levesque 2004] Ronald Brachman and Hector Levesque. *Knowledge Representation and Reasoning*. The Morgan Kaufmann Series in Artificial Intelligence. Elsevier Science, 2004. isbn: 9781558609327. doi: 10.1016/b978-1-55860-932-7.x5083-3. url: http://dx.doi.org/10.1016/B978-1-55860-932-7.X5083-3 (cit. on p. 1).

[Hansson 1989] Sven Ove Hansson. “New operators for theory change”. *Theoria* 55.2 (1989), p. 114 (cit. on p. 9).

[Hansson 1991] Sven Ove Hansson. “Belief base dynamics”. In: 1991. url: https://api.semanticscholar.org/CorpusID:141890736 (cit. on p. 7).

[Hansson 1993a] Sven Ove Hansson. “Changes of disjunctively closed bases”. *Journal of Logic, Language and Information* 2.4 (Oct. 1993), pp. 255–284. issn: 1572-9583. doi: 10.1007/bf01181682. url: http://dx.doi.org/10.1007/BF01181682 (cit. on pp. 1, 9).

[Hansson 1993b] Sven Ove Hansson. “Reversing the levi identity”. *Journal of Philosophical Logic* 22.6 (Dec. 1993), pp. 637–669. issn: 1573-0433. doi: 10.1007/bf01054039. url: http://dx.doi.org/10.1007/BF01054039 (cit. on pp. 1, 7).

[Hansson 1994] Sven Ove Hansson. "Kernel contraction". *Journal of Symbolic Logic* 59.3 (1994), pp. 845–859. doi: 10.2307/2275912 (cit. on pp. 1, 7, 8).

[Kalyanpur 2006] Aditya Kalyanpur. "Debugging and repair of owl ontologies". In: 2006. url: https://api.semanticscholar.org/CorpusID:31341190 (cit. on p. 1).

[Kazakov *et al.* 2012] Yevgeny Kazakov, Markus Krötzsch, and František Simančík. *ELK: A Reasoner for OWL EL Ontologies*. System Description. available from http://code.google.com/p/elk-reasoner/wiki/Publications. University of Oxford, 2012 (cit. on p. 32).

[Keet 2018] C. Maria Keet. "An introduction to ontology engineering". In: 2018. url: https://api.semanticscholar.org/CorpusID:115750497 (cit. on p. 1).

[Matos 2021] Vinícius Bitencourt Matos. "Pseudo-contraction operations for description logics". Master's thesis. Universidade de São Paulo, Agência USP de Gestão da Informação Acadêmica (AGUIA), 2021. doi: 10.11606/d.45.2021.tde-02092021-131750. url: http://dx.doi.org/10.11606/D.45.2021.tde-02092021-131750 (cit. on pp. 3, 10, 11).

[Matos *et al.* 2019] Vinícius Bitencourt Matos, Ricardo Guimarães, Yuri David Santos, and Renata Wassermann. "Pseudo-contractions as gentle repairs". In: *Description Logic, Theory Combination, and All That: Essays Dedicated to Franz Baader on the Occasion of His 60th Birthday*. Ed. by Carsten Lutz, Uli Sattler, Cesare Tinelli, Anni-Yasmin Turhan, and Frank Wolter. Cham: Springer International Publishing, 2019, pp. 385–403. isbn: 978-3-030-22102-7. doi: 10.1007/978-3-030-22102-7_18. url: https://doi.org/10.1007/978-3-030-22102-7_18 (cit. on pp. 1, 20, 31, 41).

[Peppas 2008] Pavlos Peppas. "Chapter 8 - belief revision". In: *Handbook of Knowledge Representation*. Foundations of artificial intelligence. Elsevier, 2008, pp. 317–360 (cit. on pp. 1, 3, 5).

[Reiter 1987] Raymond Reiter. "A theory of diagnosis from first principles". *Artificial Intelligence* 32.1 (1987), pp. 57–95. issn: 0004-3702. doi: https://doi.org/10.1016/0004-3702(87)90062-2. url: https://www.sciencedirect.com/science/article/pii/0004370287900622 (cit. on p. 19).

[Ribeiro 2010] Marcio Moretto Ribeiro. "Revisão de crenças em lógicas de descrição e em outras lógicas não clássicas". PhD thesis. Universidade de São Paulo, 2010. doi: 10.11606/t.45.2010.tde-16112010-155644. url: http://dx.doi.org/10.11606/T.45.2010.tde-16112010-155644 (cit. on p. 3).

[Santos 2016] Yuri David Santos. "Pseudo-contractions in belief revision". Master's thesis. Universidade de São Paulo, 2016. doi: 10.11606/d.45.2016.tde-08062016-105125. url: http://dx.doi.org/10.11606/D.45.2016.tde-08062016-105125 (cit. on pp. 9, 10).

[Santos *et al.* 2018] Yuri David Santos, Vinícius Bitencourt Matos, Márcio Moretto Ribeiro, and Renata Wassermann. "Partial meet pseudo-contractions". *International Journal of Approximate Reasoning* 103 (2018), pp. 11–27. issn: 0888-613X. doi: https://doi.org/10.1016/j.ijar.2018.08.006. url: https://www.sciencedirect.com/science/article/pii/S0888613X1830104X (cit. on pp. 1, 9, 11, 20, 41).

[Schlobach and Cornet 2003] 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'03. Acapulco, Mexico: Morgan Kaufmann Publishers Inc., 2003, pp. 355–360 (cit. on p. 19).

[Wassermann 2000] Renata Wassermann. "Resource-Bounded Belief Revision". ILLC Dissertation Series 2000-01. PhD thesis. Universiteit van Amsterdam, 2000. isbn: 90-5776-040-1 (cit. on p. 8).