Pushing the Box$\mathcal{EL}$ Envelope

MSc Thesis *(Afstudeerscriptie)*

written by

Amity Aharoni
(born January 12th, 1995 in Petah-Tiqwa, Israel)

under the supervision of **Prof. Benno van den Berg** and **Prof Erman Acar**, and submitted to the Examinations Board in partial fulfillment of the requirements for the degree of

MSc in Logic

at the *Universiteit van Amsterdam*.

Date of the public defense:
August 28, 2024

Members of the Thesis Committee:
Prof Maria Aloni
Prof Balder ten Cate
Prof Ronald de Haan

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

Abstract

This paper formalizes knowledge base embedding algorithms using categorical logic, focusing on box embeddings. We introduce a novel approach utilizing hyperdoctrines, a categorical construction, to analyze the relationship between a knowledge base and its embedding space. We provide a proof of the incompleteness of state-of-the-art box embedding approaches like BoxEL, and then use algebraic tools to extend box space embeddings to a novel MultiboxEL embedding approach. We establish that every $\mathcal{EL}^{++}$ knowledge base possesses a finite model and use that to show completeness of MultiboxEL with respect to $\mathcal{EL}^{++}$. We further extend our embedding to $\mathcal{ALC}$ knowledge bases. Finally, we implement and compare our new embedding strategies against state-of-the-art box embedding models, providing empirical evidence for the usefulness and limitation of our approach. These contributions collectively offer a robust, algebraic method for knowledge base embeddings, advancing the field and opening new avenues for the application of categorical logic in artificial intelligence and machine learning.

Acknowledgements

To all those important to me, our interactions form the base space from which I lift my essence; each of you is a rich fiber in the bundle that uniquely defines me.

I would like to thank my supervisors, Prof. Benno van den Berg and Prof. Erman Acar. Your work and teaching have always fueled my passion for mathematics. Beyond your excellent academic supervision, you provided me with the freedom to pursue my interests, make mistakes, grow and flourish. Without your support, I would not have been able to write a thesis I truly believe in. Your kindness reminded me that academia is not only about knowledge, but about the community formed around it.

To my committee and the staff of the ILLC, thank you for three incredible years. I deeply appreciate the care you put into your teaching, your dedication to the art of education, and the personal touches that make this master's program so special.

To my parents, Hemi and Tania Aharonee, thank you for always supporting and guiding me. Your wisdom, understanding, and encouragement have shaped me into who I am today. You taught me to see the bigger picture, plan ahead, and inspired me to be stronger. For Orit Weinberger, your light will forever shine with us.

To my best friends, Lide Grotenhuis, Marion Dugue, and Rodrigo Almeida, without you, Amsterdam would have felt empty. Thank you for always making me smile, believing in me, and teaching me to trust others. Thank you for enriching my life with countless experiences and always being there when I made mistakes.

To my siblings, Yahel Ichilov, for always caring for me and being there when

I needed you most; Itamar Aharoni, for standing by me through the toughest times and teaching me the value of perseverance and excellence; and Assaph Aharoni, for sharing the biggest laughs and joys. Above all, thank you for never giving up on me and for always having the courage to teach me new things.

To my roommate, Lo, for opening my mind and heart to others, creating the safe space that allowed me to grow, and sharing wine during times of hardship.

Lastly, I would like to thank Elia Yom Tov. You taught me never to lose faith in the power of education. I hope to use all the lessons I've learned to help others as much as you have helped me.

Introduction

In recent years, the importance of well-curated knowledge bases, that record specialized information, has become increasingly evident across various industries and domains. Such repositories serve as critical tools for the development of specialized algorithms. The process of data collection, annotation, and validation by domain experts is called knowledge base curation.

These knowledge bases are often formally documented using description logics due to their strong algorithmic properties ([Staab and Studer, 2009]). An important case study was $\mathcal{EL}$, a description logic which allows for conjunction and existential restrictions while also being tracable ([Brandt, 2004]).

In Pushing the $\mathcal{EL}$ Envelope, it has been shown that it is possible to add a bottom element and nominals (concepts that are satisfied by only one object) to the language while maintaining tractability. The new language is called $\mathcal{EL}^{++}$. $\mathcal{EL}^{++}$ has seen use in many applications including large parts of the Galen medical knowledge base ([Rector and Horrocks, 1997]) and the Gene Ontology ([Ashburner et al., 2000]).

Despite its critical importance, knowledge base curation presents numerous challenges. The sheer volume and diversity of information sources, coupled with the dynamic nature of knowledge, make the curation process complex and resource-intensive.

Knowledge base completion is the task of automating curation by inferring missing facts through the information already present in the knowledge base. One such approach is knowledge base embeddings. Knowledge base embeddings represent entities and relationships within the knowledge base as vectors in a low-dimensional metric space. By treating the distance function of the space as a similarity score between vectors, the embedding allows us to complete the knowledge base with potentially new connections between entities.

This led to the development of the state-of-the-art methods BoxEL ([Xiong et al., 2022a]) which represents concepts in $\mathcal{EL}^{++}$ as boxes (i.e., axis-aligned hyperrectangles) and roles as affine transformations. While BoxEL manages to preserve many structural properties of the knowledge base, it is unable to capture one-to-many, many-to-one, or many-to-many relations and thus is limited in its ability to faithfully represent inclusion relations in $\mathcal{EL}^{++}$. Followup papers attempted to improve BoxEL by devising representations for the $\mathcal{EL}^{++}$ constructors that faithfully preserve the underlying structure of the logic.

A reasonable requirement for these embeddings is that any inferred facts from the embeddings are correct inferences from the existing knowledge base, preventing the introduction of incorrect information. And that all possible true facts that can be inferred from the existing knowledge are captured, ensuring no relevant information is missed. These two requirements can be phrased in terms of soundness and completeness. While proving these results is crucial for ensuring the faithfulness of the embeddings and their interpretable value, most knowledge base embeddings do not lay a mathematical foundation deep enough to verify these properties.

To formalize and analyze the relationship between a knowledge base and its embedding space, we would use hyperdoctrines. Hyperdoctrines are a categorical construction that describes a relationship between first-order logics and geometric spaces ([Lawvere, 1969]). Hyperdoctrines allow us to uniquely construct the representation of the $\mathcal{EL}^{++}$ constructors within a space in which concepts are represented as subsets of a space, which also preserves all the logical structure of the underlying knowledge base. Due to these properties of hyperdoctrines, we believe them to be very useful in the discussion about knowledge-base embeddings.

The motivating example for our approach would be box embeddings of $\mathcal{EL}^{++}$ and $\mathcal{ALC}$ knowledge bases. Box embedding is a popular embedding method of $\mathcal{EL}^{++}$. In this approach, concepts are represented as hyperrectangles (or boxes) in a vector space. We would prove soundness and completeness of the box embedding representation that we construct using categorical logic and experiment with AI embedding algorithms to compare the performance and accuracy of our categorical logic-based embeddings against traditional box emebddings.

We conclude this paper with an implementation of a new embedding strategy called MultiboxEL. The MultiboxEL embedding allow us to represent $\mathcal{EL}^{++}$ as well as more expressive and non-tractable languages such as $\mathcal{ALC}$.

This thesis makes several significant contributions to the field of knowledge base embeddings:

1. **Formalization of Box Embeddings for Description Logics through Hyperdoctrines**: We provide a formal framework using hyperdoctrines to represent and reason about box embeddings for description logics, ensuring soundness and completeness. Unlike previous papers that took an experimental approach to represent logical operations, our algebraic method allows us to compute directly and uniquely what the representations of the operations should be to ensure completeness.

2. **Proof that Box Embeddings are Not Complete Against $\mathcal{EL}^{++}$ Knowledge Bases**: We show that traditional box embeddings fail to achieve completeness when applied to $\mathcal{EL}^{++}$ knowledge bases, despite the robust literature that applies them for $\mathcal{EL}^{++}$ knowledge base embeddings.

3. **Proof that Every $\mathcal{EL}^{++}$ Knowledge Base has a Finite Model**: We establish that every $\mathcal{EL}^{++}$ knowledge base possesses a finite model. Despite the wide usage of $\mathcal{EL}^{++}$, we could not find a proof of this important result.

4. **A New Strategy for Embedding $\mathcal{EL}^{++}$ Knowledge Bases Using Multiboxes**: We introduce a novel embedding approach utilizing multiboxes, which we show to be complete with respect to $\mathcal{EL}^{++}$ knowledge bases.

5. **A Strategy for Embedding $\mathcal{ALC}$ Knowledge Bases Using Multiboxes**: We extend our multibox embedding strategy to $\mathcal{ALC}$ knowledge bases, demonstrating its applicability to a broader range of description logics.

6. **Experimentation Section**: We implement our new embedding strategies using a model called MultiboxEL. We then conduct experiments comparing the performance and accuracy of our embeddings against classical models in the literature, providing empirical evidence of the effectiveness of our approach.

Related Works

Hyperdoctrines

The use of lattices to interpret logical systems marked a significant milestone in the history of logic. It offered an algebraic representation of the operations of propositional calculus [Birkhoff, 1967].

Hyperdoctrines, introduced by Lawvere [Lawvere, 1969], showed that through the categorical notion of adjoint functor, it is possible to algebraically represent quantifiers over lattices, thus allowing for an algebraic representation of first-order logic. Hyperdoctrines require the lattices to be generated above a category of contexts.

Knowledge Base Embeddings

Early works in knowledge base embeddings embed entities and relations between them as vectors ([Bordes et al., 2013]). The measure of how much relation $\mathbf{r}$ holds between two entities $\mathbf{a}, \mathbf{b}$ is represented as a translation $\mathbf{b} - (\mathbf{a} + \mathbf{r})$. While such models manage to predict the relation between entities well, they do not represent concepts and subsumption relation and thus are incapable of representing ontological data and complex logical queries.

Other models suggested to use convex spaces such as balls to represent concepts ([Gutiérrez-Basulto and Schockaert, 2018], [Kulmanov et al., 2019a]). [Xiong et al., 2022b] proposed to use boxes because they are a computationally efficient representation, while also capturing faithfully the meaning of conjunction as intersection of two boxes and thus introduced the model BoxEL. Following the original TransE, roles were represented in BoxEL as affine transformations. [Jackermeier et al., 2024a] managed to represent both concepts and roles as boxes with the model BoxSquaredEL. While the researchers managed to show that their system is sound with respect to $\mathcal{EL}^{++}$[1], they have not considered completeness. Moreover, since their representation does not rely on algebraic methods, they cannot express complicated logical queries with their approach, nor extend it to new description logics. We show in this paper that their method is in fact not complete with respect to $\mathcal{EL}^{++}$ and propose a representation that manages to express complicated concepts and is extendable.

[1] We say an embedding is sound if when the loss measure of the embedding with respect to a knowledge base is zero, then the embedding satisfies the knowledge base

Part I

Preliminaries

Chapter 1

Description Logic

Knowledge representation is a field of study in artificial intelligence that focuses on how knowledge can be represented, organized, and stored. A system of knowledge representation needs to be sufficiently expressive, in order to capture as much information of the domain subject as possible.

A knowledge representation also needs to be computationally efficient. That is, given a set of facts, and a query about the facts, the problem of solving the query should have low complexity.

First-order logic is often considered as an initial framework for knowledge representation. It provides a formal language for expressing statements and rules about objects, relationships, and properties in a domain.

However, despite its expressiveness, first order logic is not sufficient for many applications due to the fact that first order logic is not decidable [Avron et al., 2008]. Knowledge representation focuses on decidable fragments of first order logic. Description logics (DLs) are a family of first order languages with decidable properties.

In the following sections, we describe the syntax and semantics of the description logics analyzed in this thesis. This chapter relies on the construction found in [van Harmelen et al., 2007].

1.1 $\mathcal{ALC}$ Syntax and Semantics

We begin by introducing the syntax and semantics of one of the most expressive DLs, namely the Attributive Concept Language with Complements DL ($\mathcal{ALC}$).

1.1.1 Syntax

The syntax of a DL is defined by a signature and grammar. The signature defines the set of primitive names of the DL (e.g., concept names, role names, and individual names). The grammar defines how to combine primitive names to form complex names (e.g., concepts and roles). The set of complex names is called the set of concepts and roles of the DL.

Definition 1.1 (Signature)**.** *Given a description logic (DL), a* signature Σ *is a triple* $\Sigma = (N_{\mathcal{C}}, N_{\mathcal{R}}, \mathcal{O})$*, where* $N_{\mathcal{C}}$*,* $N_{\mathcal{R}}$*, and* $\mathcal{O}$ *are pairwise disjoint sets of* concept names, roles, *and* individuals, *respectively.*

A description logic is defined by the set of constructors it supports. The constructors of $\mathcal{ALC}$ are $\exists$, $\forall$, $\sqcap$, $\sqcup$, $\neg$, $\top$, and $\bot$.

Definition 1.2 ($\mathcal{ACL}$ Role)**.** *Given a signature* Σ *and a natural number* n*, a* $\mathcal{ALC}$-role R *is defined as a chain* $R_1 \circ \cdots \circ R_n$ *where* $R_1, \ldots, R_n \in N_{\mathcal{R}}$*.*

The set of $\mathcal{ALC}$*-roles over a signature* Σ *is denoted by* $\mathcal{R}$ *and is defined as the smallest set that contains* $N_{\mathcal{R}} \subseteq \mathcal{C}$ *and every* $\mathcal{ALC}$*-role.*

Definition 1.3 ($\mathcal{ACL}$ Concept)**.** *Given a signature* Σ*, a* $\mathcal{ALC}$-concept C *is defined recursively by the following grammar:*

$$C ::= A \mid \top \mid \bot \mid D \sqcap E \mid D \sqcup E \mid \neg D \mid \exists R.D \mid \forall R.D$$

where $A \in N_{\mathcal{C}}$*,* $R \in \mathcal{R}$ *and* D, E *are previously defined* $\mathcal{ALC}$*-concepts. The set of* $\mathcal{ALC}$*-concepts over a signature* Σ *is denoted by* $\mathcal{C}$ *and is defined as the smallest set that contains* $N_{\mathcal{C}} \subseteq \mathcal{C}$ *and every concept* C *that can be recursively constructed from the primitive concepts in* $N_{\mathcal{C}}$ *using the above grammar.*

1.1.2 Semantics

In logic, the semantics of a language describe how to interpret the language in some domain. The semantics of a DL is defined as a set of elements that can be used to interpret the primitive names of the DL. The interpretation is a mapping from the primitive names of the DL to elements of the fixed domain. The interpretation of a complex name is defined in terms of the interpretation of its subnames. The standard semantics of $\mathcal{ALC}$ treats concepts as subsets of the fixed domain and roles as binary relations over the fixed domain.

Definition 1.4 (Interpretation)**.** *Given a signature* Σ*, an* interpretation $\mathcal{I}$ *is a triple* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}_{\mathcal{C}}, \cdot^{\mathcal{I}}_{\mathcal{R}}, \cdot^{\mathcal{I}}_{\mathcal{O}})$*, where* $\Delta^{\mathcal{I}}$ *is a non-empty set called the* domain *of* $\mathcal{I}$*,*

And we have interpretation maps

$$\begin{aligned} &\cdot^{\mathcal{I}}_{\mathcal{C}} : \mathcal{C} \to \mathcal{P}\Delta^{\mathcal{I}} \\ &\cdot^{\mathcal{I}}_{\mathcal{O}} : \mathcal{O} \to \Delta^{\mathcal{I}} \\ &\cdot^{\mathcal{I}}_{\mathcal{R}} : \mathcal{R} \to \mathcal{P}(\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}) \end{aligned}$$

such that $\cdot^{\mathcal{I}}_{\mathcal{C}}$ *maps each concept* $C \in \mathcal{C}$ *to a subset* $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$*, and* $\cdot^{\mathcal{I}}_{\mathcal{R}}$ *maps each role names* $R_1, \ldots, R_n \in N_{\mathcal{R}}$ *to a binary relation* $R_1^{\mathcal{I}}, \ldots, R_n^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$*.*

The interpretation function $\cdot^{\mathcal{I}}$ *must satisfies the following conditions:*

- $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$ *and* $\bot^{\mathcal{I}} = \emptyset$*.*

- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$ *and* $(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$.
- $(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$.
- $(R_1 \circ \cdots \circ R_n)^{\mathcal{I}} = R_1^{\mathcal{I}} \circ \cdots \circ R_n^{\mathcal{I}}$
- $(\exists R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}}.(x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$.
- $(\forall R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \forall y \in \Delta^{\mathcal{I}}.(x,y) \in R^{\mathcal{I}} \implies y \in C^{\mathcal{I}}\}$.

1.1.3 Knowledge Base

Description logics are used to represent knowledge about a domain of interest (e.g., a medical domain). The knowledge about a domain of interest is represented by a knowledge base. A knowledge base is a set of axioms where an axiom is a statement that describes a relation between concepts, roles, and individuals. The semantics of a knowledge base is defined in terms of the semantics of its axioms.

Definition 1.5 (TBox and Subsumption)**.** *Given a signature* Σ*, a* TBox $\mathcal{T}$ *is a finite set of* subsumption axioms *of the form* $C \sqsubseteq D$ *or* $R \sqsubseteq K$*, where* $C, D \in \mathcal{C}$ *and* $R, K \in \mathcal{R}$.

Notation 1.6 (Definition)**.** *Given a signature* Σ*, a* definition, *denoted as* $C \equiv D$*, is an abbreviation for the subsumption axioms* $C \sqsubseteq D$ *and* $D \sqsubseteq C$.

Definition 1.7 (ABox and Assertion)**.** *Given a signature* Σ*, an* ABox $\mathcal{A}$ *is a finite set of* assertion axioms *of the form* $C(a)$ *or* $R(a,b)$*, where* $C \in \mathcal{C}$*,* $R \in \mathcal{R}$*, and* $a, b \in \mathcal{O}$.

Definition 1.8 (Satisfiability)**.** *Given an interpretation* $\mathcal{I}$*, we say that*

- $\mathcal{I}$ satisfies *a concept* $C \in \mathcal{C}$ *if* $C^{\mathcal{I}} \neq \emptyset$.
- $\mathcal{I}$ satisfies *a role* $R \in \mathcal{R}$ *if* $R^{\mathcal{I}} \neq \emptyset$.
- $\mathcal{I}$ satisfies *a subsumption axiom* $C \sqsubseteq D$ *if* $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ *and* $R \sqsubseteq K$ *if* $R^{\mathcal{I}} \subseteq K^{\mathcal{I}}$.
- $\mathcal{I}$ satisfies *an assertion axiom* $C(a)$ *if* $a^{\mathcal{I}} \in C^{\mathcal{I}}$.
- $\mathcal{I}$ satisfies *an assertion axiom* $R(a,b)$ *if* $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$.
- $\mathcal{I}$ satisfies *a TBox* $\mathcal{T}$ *if* $\mathcal{I}$ *satisfies every subsumption axiom in* $\mathcal{T}$.
- $\mathcal{I}$ satisfies *an ABox* $\mathcal{A}$ *if* $\mathcal{I}$ *satisfies every assertion axiom in* $\mathcal{A}$.

Definition 1.9 (Knowledge Base)**.** *Given a signature* Σ*, a* knowledge base $\mathcal{K}$ *is a pair* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$*, where* $\mathcal{T}$ *is a TBox and* $\mathcal{A}$ *is an ABox.*

Definition 1.10 (Model)**.** *Given a knowledge base* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ *and an interpretation* $\mathcal{I}$*, we say that* $\mathcal{I}$ *is a* model *of* $\mathcal{K}$ *if* $\mathcal{I}$ *satisfies* $\mathcal{T}$ *and* $\mathcal{A}$.

We say that a knowledge base $\mathcal{K}$ *is consistent if it has a model.*

Notation 1.11 (Satisfiability)**.** *We will write* $\mathcal{I} \models \mathcal{L}$ *if* $\mathcal{I}$ *satisfies* $\mathcal{L}$.

Under this semantics, a query q is entailed by a knowledge base $\mathcal{K}$ if q is true in all models of $\mathcal{K}$. We denote this by $\mathcal{K} \vdash q$.

1.2 $\mathcal{EL}^{++}$ Syntax and Semantics

For most applications, $\mathcal{ALC}$ is too computationally expensive to be used in practice. For example, the subsumption problem in $\mathcal{ALC}$ is PSPACE-complete [Schmidt-Schauß and Smolka, 1991]. To address this issue, a number of tractable DLs have been proposed.

One of the most popular tractable DLs is the $\mathcal{EL}$ family of DLs. The $\mathcal{EL}$ family of DLs is a subset of the $\mathcal{ALC}$ family of DLs that contains only the constructors $\exists$, $\sqcap$ and $\top$.

Definition 1.12 ($\mathcal{EL}$ Role)**.** *$\mathcal{EL}$-roles are the same as $\mathcal{ALC}$ roles.*

Definition 1.13 ($\mathcal{EL}$ Concept)**.** *Given a signature Σ, an $\mathcal{EL}$-*concept *C is defined recursively by the following grammar:*

$$C ::= A \mid \top \mid D \sqcap E \mid \exists R.D$$

where $A \in N_{\mathcal{C}}$, $R \in \mathcal{R}$ and D, E are previously defined $\mathcal{EL}$-concepts. The set of $\mathcal{EL}$-concepts over a signature Σ is denoted by $\mathcal{C}$ and is defined as the smallest set that contains $N_{\mathcal{C}} \subseteq \mathcal{C}$ and every concept C that can be recursively constructed from the primitive concepts in $N_{\mathcal{C}}$ using the above grammar.

Interpretation of $\mathcal{EL}$-concepts is similar to the interpretation of $\mathcal{ALC}$-concepts. The main advantage of $\mathcal{EL}$ is that subsumption in $\mathcal{EL}$ is polynomial time [Baader, 2003]. In [Baader et al., 2005] it was shown that adding the constructor $\top$ and nominality does not increase the complexity of subsumption. The resulting DL is called $\mathcal{EL}^{++}$.

Definition 1.14 ($\mathcal{EL}^{++}$ Concept)**.** *Given a signature Σ, an $\mathcal{EL}^{++}$-*concept *C is defined recursively by the following grammar:*

$$C ::= A \mid \top \mid \bot \mid D \sqcap E \mid \exists R.D \mid \{a\}$$

where $A \in N_{\mathcal{C}}$, $a \in \mathcal{O}$, $R \in \mathcal{R}$ and D, E are previously defined $\mathcal{EL}^{++}$-concepts. The set of $\mathcal{EL}^{++}$-concepts over a signature Σ is denoted by $\mathcal{C}$ and is defined as the smallest set that contains $N_{\mathcal{C}} \subseteq \mathcal{C}$ and every concept C that can be recursively constructed from the primitive concepts in $N_{\mathcal{C}}$ using the above grammar.

Note that the inclusion of $\{a\}$ as concepts allows us to form TBoxes such as $C \sqsubseteq \{a\}$. Such constructor is called nominal.

One powerful result for $\mathcal{EL}^{++}$ is the normalization theorem:

Definition 1.15 (Basic Concept)**.** *We define the set of basic concepts as* BC $= \{a_i\}_{a_i \in \mathcal{O}} \cup \mathcal{N}_{\mathcal{C}} \cup \{\top, \bot\}$.

Lemma 1.16 ([Baader et al., 2005])**.** *Given a TBox $\mathcal{T}$, we can generate (in linear time) a normalized TBox $\mathcal{T}'$ of $\mathcal{T}$ such that all subsumptions in it are of the form*

$$\begin{aligned} C_1 &\sqsubseteq D, & C_1 &\sqsubseteq \exists R.C_2 \\ C_1 \sqcap C_2 &\sqsubseteq D, & \exists R.C_1 &\sqsubseteq D \\ R &\sqsubseteq S, & R_1 \circ R_2 &\sqsubseteq S \end{aligned}$$

where $C_1, C_2 \in \mathrm{BC} \setminus \{\bot\}$, $D \in \mathrm{BC} \setminus \{\top\}$ *and* $R, R_1, R_2, S \in N_{\mathcal{R}}$ *and where every model of* $\mathcal{T}'$ *is a model of* $\mathcal{T}$ *and every model of* $\mathcal{T}$ *can be extended to a model of* $\mathcal{T}'$.

Given any knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, *we can generate a normalized knowledge base* $\mathcal{K}' = (\mathcal{T}', \mathcal{A}')$ *of* $\mathcal{K}$ *where* $\mathcal{T}'$ *is the normalized form of* $\mathcal{T}$, *every assertion* $a : \{b\} \in \mathcal{A}$ *is removed from* $\mathcal{A}'$ *and added as a subsumption* $a \sqsubseteq \{b\} \in \mathcal{T}'$. *We have that every model of* $\mathcal{K}'$ *is a model of* $\mathcal{K}$ *and every model of* $\mathcal{K}$ *can be extended to a model of* $\mathcal{K}'$.

Chapter 2

Knowledge Base Embeddings

Neuro symbolic AI is a field of research that studies the application of symbolic representations of knowledge in neural network algorithms. In this chapter, we explain what a deep learning algorithm is and how knowledge bases can be used in the field. We begin by exploring what a learning algorithm is, then describe the particular class of neural network algorithms and finish with the application of such algorithms called knowledge base embedding.

Supervised learning algorithms attempt to locate function $f : A \to B$ that is most optimal for solving a certain task from a list of functions indexed by a set P of parameters $I : P \times A \to B$, given a set of points called the dataset $\mathbb{D} := \{(a_1, b_1), \ldots\}$. We call $b_1, \ldots$ the labels of the dataset. In neural network algorithms, P is some hyperplane $\mathbb{R}^n$. The optimal solution can be characterized through a distance function called the loss function $L : A \times B \to \mathbb{R}$ that measures the distance of a particular function on some input $I(p, -)(a)$ to the desired output b.

The most straightforward approach for finding the optimal solution is to compute the partial derivative over the set of parameters and identify the critical points. That is, solving the equation

$$\nabla_P L(I(-), -) = \vec{0} \tag{2.1}$$

However, it is not always feasible to solve 2.1.

Instead, neural networks converge to a local critical point through a stochastic gradient descent algorithm. Given a constant known as the learning rate $\eta \in \mathbb{R}$ and some $\varepsilon \in \mathbb{R}$, the gradient descent algorithm is:

Algorithm 1 Stochastic Gradient Descent

```
Randomly sample a p ∈ P
while L(I(−),−) < ε do
    Sample a subset D ⊆ 𝔻.
    for (a,b) ∈ D do
        Compute the loss L(I(a,−),b).
        Compute the gradient ∇L(I(p,a),b).
        Update the parameters p ← p − η∇L(I(p,a),b).
    end for
end while
```

It is important to note that the gradient descent algorithm usually does not reach the critical point. Rather, it approximates it sufficiently well.

The gradient descent process is called the learning algorithm. The computation of gradient is called a backward pass and the computation of the loss function is called the forward pass.

Parameters that remain fixed throughout the learning process such as η and ε are called hyper parameters.

2.1 The Neural Network Architecture

Definition 2.1 (Neural Network Architecture)**.** *A neural network architecture is a tuple* $\langle G, \{\sigma_L\}_{L\leq N\in\mathbb{N}}\rangle$ *such that*

- *G is a finite directed graph $\langle V, R\rangle$ where $V := \bigsqcup_{L\leq N+1} V_L$ and for any $(v,w) \in R$, there exists some $L \leq N$ such that $v \in V_L$ and $w \in V_L + 1$.*
- *For any L, we have that σ_L is a non linear function $\sigma_L : \mathbb{R} \to \mathbb{R}$.*

We call V_1 and V_{N+1} the input and output layers of the neural network. For any L, we call $|V_L|$ the width of the L^{th} layer and σ_L the activation function. We call $N+1$ the depth of the network.

The neural network architecture represents the I function in the previous section.

Definition 2.2 (Neural Network)**.** *Given a neural network architecture* $\langle G, \{\sigma_L\}_{L\leq N\in\mathbb{N}}\rangle$ *and a compact space $K \subseteq \mathbb{R}^d$ a neural network is a function $f : K \to \mathbb{R}^{|V_{L+1}|}$ such that*

1. *There exists a set of linear functions $\{T_L(x) := W_L(x) + b_L\}_{L\in[1,N+1]}$ where the dimensions of W_1 are $d \times |V_1|$.*
2. *The dimensions of W_{L+1} are $|V_L| \times |V_{L+1}|$.*
3. *There exists some enumeration $V_L := \{v_{1,L}, \ldots, v_{n,L}\}, V_{L+1} := V_L := \{v_{1,L+1}, \ldots, v_{m,L+1}\}$ such that if $(v_{q,L}, v_{p,L+1}) \notin R$ then $W_{p,q} = 0$.*

4. *We have that*
$$f(x) = (T_{N+1} \circ \sigma_N \odot T_N \circ \cdots \circ \sigma_1 \odot T_1)(x)$$
where $\odot$ represents component wise composition.

Given a layer L, W_L is called the weights of layer L and b_L is called the biases of layer L.

The biases are hyper parameters of the algorithm and $P = W_1 \times \cdots \times W_{N+1}$.

One important reason that neural networks are very useful is the fact that they can limit any continuous function.

Theorem 2.3 (Universal approximation theorem, [Haykin, 1999]). *Given a function $\sigma : \mathbb{R} \to \mathbb{R}$, we have that σ is non-linear if and only if for every natural numbers d, m, compact space $K \subseteq \mathbb{R}^d$, continuous map $f : K \to \mathbb{R}^m$ and $\varepsilon > 0$, there exists a neural network $g : K \to \mathbb{R}^m$ with architecture $\langle G, \{\sigma\}\rangle$ such that*

$$\sup_{x \in K} \|f(x) - g(x)\| < \varepsilon$$

Notice that theorem 2.3 tells us that for any such map f, there exists a neural network architecture of depth 3 that satisfies the condition. However, it does not specify how to find such a function. Furthermore, it has been empirically observed that deep and narrow neural networks tend to be more optimal for finding an approximating function better than wide and shallow neural networks [Nguyen et al., 2021]. Finding a good architecture relies more on empirical rather than theoretical work.

2.2 Knowledge Base Embedding

Given a signature Σ and a given knowledge base $\mathcal{K}$, a knowledge representation learning task is a training task for finding interpretations of the description logic that approximate the satisfaction of $\mathcal{K}$. That is, the task is to produce an interpretations $(\mathbb{R}^n, \cdot^{\mathcal{I}}_{\mathcal{C}}, \cdot^{\mathcal{I}}_{\mathcal{D}})$ where we interpret the subsumption relation using some function $\sqsubset: \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ and we measure how effective is this interpretation using some cost function $L : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ that measures how well does the interpretation satisfies the subsumption relation between concepts in $\mathcal{K}$. The interpretations $\mathcal{I}$ are called knowledge base embeddings. Usually, a training task searches for such an embedding under a fixed domain. That is under a fixed dimension of $\mathbb{R}$.

The reason we are interested in such tasks is that storing complete knowledge bases and computing the subsumption relation between concepts within them can be very computationally and memory expensive. Knowledge base embeddings allow us to find a computationally cheap representation that captures the core structure of the graph.

Chapter 3

Categorical Logic

Algebraic and categorical logic are rich languages that allow us to mathematically formulate what it means for a given embedding to capture the structure of a knowledge base and to find an optimal representation of different concepts from the knowledge base that respect the different description logic operators. We would begin this section by covering the algebraic logic structures and then move to describing them in the language of category theory.

3.1 Algebraic Logic

Algebraic logic is a mathematical description of logic that uses posets to represent the relation between formulas.

Definition 3.1 (Poset)**.** *A poset is a pair* $(A, \leq)$ *where* A *is a set and* $\leq$ *is a relation such that*

- *For any* $p \in A$*, we have that* $p \leq p$ *(reflexivity).*
- *For any* $p, q \in A$*, if* $p \leq q$ *and* $q \leq p$ *then* $p = q$ *(antisymetry).*
- *For* $p, q, r \in A$*, if* $p \leq q$ *and* $q \leq r$ *then* $p \leq r$ *(transitivity).*

Definition 3.2 (Bounded poset)**.** *A poset is bounded if and only if there exists two elements* 0 *and* 1 *in* A *such that for any* $p \in A$ *we have that* $0 \leq p$ *and* $p \leq 1$*.*

Example 3.3 (Propositional logic)**.** *Given a set of propositions* P*, the provability relation* $\vdash$ *(that is* $p \vdash q$ *if assuming* p *then* q *is true) forms a poset over* P*. Moreover, the propositions* $\bot$ *and* $\top$ *bound the poset.*

The $\vdash$ relation is sufficient all the logic operators by describing their relation to provability

Definition 3.4 (Meets and join). *For any two elements p,q of a poset $(P,\le)$, we define the meet $p \wedge q$ and the join $p \vee q$ as the elements such that for any $t \in P$,*

$$t \le p \wedge q \iff t \le p \text{ and } t \le q$$
$$p \vee q \le t \iff p \le t \text{ and } q \le t$$

Given a subset $S \subseteq P$, we define the arbitrary meet $\bigwedge S$ and arbitrary join $\bigvee S$ as the elements such that

$$t \le \bigwedge S \iff \forall p \in S, t \le p$$
$$\bigvee S \le t \iff \forall p \in S, p \le t$$

Definition 3.5 (Implication). *Given a poset $(P,\le)$ and two elements $p,q \in P$, we define the implication $p \to q$ as an object such that for any $t \in P$,*

$$t \le p \to q \iff t \wedge p \le q$$

If $\bigvee\{t : t \wedge p \le q\}$ exists, then it is equal to $p \to q$.

Definition 3.6 (Negation). *Given an element of a poset $p \in P$, we define negation $\neg p$ as $p \to \bot$*

Definition 3.7 (Lattices). *Given a poset $(P,\le)$, we say that*

- *$(P,\le)$ is a join semi-lattice if every pair of elements p,q have a join element $p \vee q$. Likewise, a meet semi-lattice if every pair has a meet element $p \wedge q$.*
- *$(P,\le)$ is a lattice if it is both a meet and a join semi-lattice.*
- *$(P,\le)$ is a complete join semi-lattice if it has all arbitrary joins. That is, for every subset S we have an element $\bigvee S$. Likewise, $(P,\le)$ is a meet semi-lattice if it has all arbitrary meets.*
- *$(P,\le)$ is a complete lattice if it is both a complete join semi-lattice and a complete meet semi-lattice.*

Definition 3.8 (Heyting and Boolean algebras). *A Heyting algebra is a structure $(H,\le,\wedge,\vee,\to,\bot,\top)$ such that $(H,\le,\wedge,\vee,\bot,\top)$ is a bounded lattice and $\to$ is an implication operator.*

A Boolean algebra is a structure $(B,\le,\wedge,\vee,\to,\bot,\top)$ such that $(B,\wedge,\vee,\Rightarrow,\bot,\top)$ is a Heyting algebra and $\to$ satisfies the law of excluded middle $a \vee \neg a = \top$ for every $a \in B$.

Definition 3.9 (Atomic element). *Given a bounded poset P, an atom a is an element such that $a \neq \bot$ and for any other element p, if $\bot < p \le a$ then $a \le p$.*

3.2 Categories

Definition 3.10 (Category). *Given a collection of objects* $\mathcal{C}_0$ *and a collection of maps* $\mathcal{C}_1$*, a category* $\mathcal{C}$ *is a tuple* $(\mathcal{C}_0, \mathcal{C}_1, \circ)$ *where*

- *Each map* $f \in \mathcal{C}_\infty$ *is assigned two objects* $A, B \in \mathcal{C}_0$ *called the domain and codomain respectively. We denote maps as* $f : A \to B$
- $\circ : \mathcal{C}_1 \times \mathcal{C}_1 \to \mathcal{C}_1$ *is a partial operation such that for every* $f : A \to B$ *and* $g : B \to C$*, we have a composite map* $g \circ f : A \to C$ *(composition).*
- *For every* $A \in \mathcal{C}_0$*, there exists an identity map* $\mathbf{1}_A : A \to A$ *such that for every other map* $f : A \to B$*, we have* $f \circ \mathbf{1}_A = f$ *and* $\mathbf{1}_B \circ f = f$*.*
- *For every* $f : A \to B$ *and* $g : B \to C$ *and* $h : C \to D$*, we have* $h \circ (g \circ f) = (h \circ g) \circ f$ *(associativity).*

Given a category $\mathcal{C}$ and two objects $A, B \in \mathcal{C}$, we often denote the set of all maps between A and B as $\mathcal{C}(A, B)$.

Definition 3.11 (Isomorphism). *Given a category* $\mathcal{C}$ *and two objects* A, B *in* $\mathcal{C}$*, we say that* A *is isomorphic to* B *(denoted by* $A \cong B$*) if there exists two arrows* $f : A \to B$ *and* $g : B \to A$ *such that* $f \circ g = \mathbf{1}_B$ *and* $g \circ f = \mathbf{1}_A$*. It is easy to show that* g *is unique. We call* g *the inverse of* f *and denote it by* f^{-1}*.*

Example 3.12 (Category of sets). *The category of sets* **Set** *is a category such that*

- *The objects of* **Set** *are the sets.*
- *The morphisms of* **Set** *are the functions between sets.*
- *The composition of two maps* $f : A \to B$ *and* $g : B \to C$ *is the regular composition of functions* $g \circ f : A \to C$*.*
- *The identity map of a set* A *is the map* $\mathbf{1}_A : A \to A$ *such that for any* $a \in A$*,* $\mathbf{1}_A(a) = a$
- *Two sets* A *and* B *are isomorphic if and only if there exists a bijection between the two of them.*

Definition 3.13 (Subcategory). *Given a category* $\mathcal{C}$*, a subcategory* $\mathcal{D}$ *of* $\mathcal{C}$ *is a pair of subsets* $\mathcal{D}_0 \subseteq \mathcal{C}_0$ *and* $\mathcal{D}_1 \subseteq \mathcal{C}_1$ *such that*

- *For every* $A \in \mathcal{D}_0$*,* $\mathbf{1}_A \in \mathcal{D}_1$
- *For every* $f : A \to B \in \mathcal{D}_1$*,* $A, B \in \mathcal{D}_0$
- *For every* $f : A \to B \in \mathcal{D}_1$ *and* $g : B \to C \in \mathcal{D}_1$*,* $g \circ f \in \mathcal{D}_1$

Definition 3.14 (Initial and Terminal Objects). *Given a category* $\mathcal{C}$*, an object* $A \in \mathcal{C}_0$ *is called initial if for every* $B \in \mathcal{C}_0$*, there exists a unique map* $f : A \to B$*. An object* $A \in \mathcal{C}_0$ *is called terminal if for every* $B \in \mathcal{C}_0$*, there exists a unique map* $f : B \to A$*. We denote these objects by* $\bot$ *and* $\top$ *respectively.*

Example 3.15 (The initial and terminal objects in **Set**). *The empty set $\varnothing$ is an initial object in the category* **Set**. *Every singleton set $\{*\}$ is a terminal object in the category* **Set**.

Definition 3.16 (Binary Products and coproducts). *Given a category $\mathcal{C}$ and two objects $A, B \in \mathcal{C}_0$, a binary product of A and B is an object $A \times B$ equipped with two maps $\pi_A : A \times B \to A$ and $\pi_B : A \times B \to B$ such that for any object $C \in \mathcal{C}_0$ and two maps $f : C \to A$ and $g : C \to B$, there exists a unique map $(f, g) : C \to A \times B$ such that*

$$\pi_1 \circ (f, g) = f \text{ and } \pi_2 \circ (f, g) = g$$

A binary coproduct of A and B is an object $A + B$ and two maps $i_A : A \to A + B$ and $i_B : B \to A + B$ such that for any object $C \in \mathcal{C}_0$ and two maps $f : A \to C$ and $g : B \to C$, there exists a unique map $f + g : A + B \to C$ such that

$$f + g \circ i_A = f \text{ and } f + g \circ i_B = g$$

Example 3.17 (Products and Coproducts in **Set**). *In* **Set**, *the products are the cartesian products and the coproducts are the disjoint unions. Given two sets A, B, the maps π_A and π_B are the projection maps. The maps i_A and i_B are the injections.*

Given a set C with maps $f : C \to A$ and $g : C \to B$, the unique map $(f, g) : C \to A \times B$ is the map $(f, g) : c \mapsto (f(c), g(c))$. Given a set C with maps $f : A \to C$ and $g : B \to C$, the unique map $f + g : A \sqcup B \to C$ is $f + g : a \mapsto \begin{cases} f(a) \text{ if } a \in A \\ g(a) \text{ if } a \in B \end{cases}$.

For a proof see [Riehl, 2017].

Remark 3.18 (Duality). *Notice that the definition of a coproduct is the same as the definition of a product but with the arrows reversed. When this is the case, we say that the object is dual to the other object, Often times duality allows us to prove theorems about one object by proving the dual theorem about the dual object (i.e. by reversing the arrows).*

The dual of a category $\mathcal{C}$ is called the opposite category and denoted by $\mathcal{C}^{op}$. It is defined as the category with the arrows of $\mathcal{C}$ reversed.

Definition 3.19 (Product and coproduct). *Given a collection of objects $\{A_i\}_{i \in I}$, a product $\times_{i \in I} A_i$ is an object alongside a collection of arrows $\{\pi_i : \times_{i \in I} A_i \to A_i\}_{i \in I}$ such that for any object $C \in \mathcal{C}_0$ and collection of maps $\{f_i : C \to A_i\}_{i \in I}$ there exists a unique map $h : C \to \times_{i \in I} A_i$ such that for any $i \in I$ we have that $\pi_i h = f_i$.*

Lemma 3.20 (Binary products and terminal object imply finite products, [Riehl, 2017]). *If a category has all binary products and a terminal object, it has all finite products. That is, if the category has a terminal object and for every two objects A, B there exists a product $A \times B$, it has all products of sets of objects $\{A_i\}_{i \in I}$ where I is finite.*

Dually, a category has all coproducts if it has an initial object and all coproducts.

Definition 3.21 (Exponential object). *Given a category $\mathcal{C}$ and two objects $A, B \in \mathcal{C}_0$, an exponential of A and B is an object B^A and a map $\epsilon : B^A \times A \to B$ such that for* *any object $C \in \mathcal{C}_0$ and a map $f : C \times A \to B$, there exists a unique map $g : C \to B^A$ such that*

$$\epsilon \circ (g, \mathbf{1}_A) = f$$

Definition 3.22 (Cartesian closed category)**.** *We call a category $\mathcal{C}$ Cartesian closed if it has a terminal object and all binary products and exponential objects. That is, for any two objects A, B in $\mathcal{C}$ we have a product object $A \times B$ and an exponential object B^A.*

Example 3.23 (Heyting algebra as a category)**.** *Every Heyting algebra can be considered as a Cartesian closed category that furthermore has an initial object and is closed under coproducts. That is, for every pair of objects A, B we have an object $A + B$. The interpretation is*

Initial Object $\bot$	Lower bound $\bot$
Terminal Object $\top$	Upper bound $\top$
Product $A \times B$	Meet $p \wedge q$
Coproduct $A + B$	Join $p \vee q$
Exponential B^A	Implication $p \to q$

Remark 3.24 (Lattices as categories)**.** *Likewise, every semi-lattice, bounded poset and lattice can be considered as a category with a suitable structure. E.g. a meet semi-lattice has all binary products, a bounded poset has terminal and initial objects, a complete lattice has arbitrary products and coproducts, etc.*

3.2.1 Limits and Colimits

We would not provide a detailed description of limits and colimits in this chapter, but due to their importance, we would offer some key characterizations of these objects.

Definition 3.25 (Pullback and pushout)**.** *Given a category $\mathcal{C}$ and two arrows $f : A \to C$ and $g : B \to C$, the pullback of f and g is an object $A \times_C B$ and two arrows $p_1 : A \times_C B \to A$ and $p_2 : A \times_C B \to B$ satisfying*

- $f \circ p_1 = g \circ p_2$.
- *For any object X and arrows $h_1 : X \to A$ and $h_2 : X \to B$ such that $f \circ h_1 = g \circ h_2$, there exists a unique arrow $u : X \to A \times_C B$ such that $p_1 \circ u = h_1$ and $p_2 \circ u = h_2$.*

The pushout is the dual object to the pullback. That is, given two arrows $f : A \to B$ and $g : A \to C$, the pushout of f and g is an object $B +_A C$ and two arrows $i_1 : B \to B +_A C$ and $i_2 : C \to B +_A C$ satisfying

- $i_1 \circ f = i_2 \circ g$.

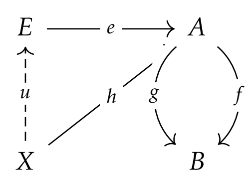
- *For any object X and arrows $h_1 : A \to B$ and $h_2 : A \to C$ such that $h_1 \circ f = h_2 \circ g$, there exists a unique arrow $u : B +_A C \to X$ such that $u \circ i_1 = h_1$ and $cu \circ i_2 = h_2$.*

Example 3.26 (Pullbacks and pushouts in the category of sets)**.** *Given two functions $f : A \to C$ and $g : B \to B$, the pullbacks in* **Set** *is the set $A \times_C B = \{(a,b) : f(a) = g(b)\}$ with the projection maps.*

Given two functions $f : A \to B$ and $g : A \to C$, define an equivalence relation $\sim$ on the disjoint union $b \sqcup C$ as the finest equivalence relation such that $f(a) \sim g(a)$ for all $a \in A$.

The pushout is the quotient $B \sqcup C/\sim$ with $i_1 : b \mapsto [a]$ and $i_2 : c \mapsto [c]$.

Definition 3.27 (Equalizer and co-equalizers)**.** *Given a category $\mathcal{C}$ and two arrows $f, g : A \to B$ in $\mathcal{C}$, the equalizer of f and g is an object E and an arrow $e : E \to A$ such that $f \circ e = g \circ e$ and for any object X and arrow $h : X \to A$ such that $f \circ h = g \circ h$, there exists a unique arrow $u : X \to E$ such that $e \circ u = h$.*



The co-equalizers are the dual object.

Example 3.28 (Equalizers and coequalizers in **Set**)**.** *Given two functions $f : A \to B$ and $g : A \to B$, the equalizer in* **Set** *is the set $\{a \in A : f(a) = g(a)\}$. The unique map u is the corestriction of h to E - $h \restriction^E : x \mapsto h(x)$.*

Define the equivalence relation $\sim$ on B as the smallest equivalence relation such that for any $a \in A$ we have that $f(a) \sim g(a)$.

The coequalizer is then the quotient $B/\sim$ alongside the quotient map $q : B \to B/\sim$. Given a set X with map $q' : B \to X$ such that $q'f = q'g$, the unique map $u : B/\sim \to X$ is q' itself.

For a proof see [Riehl, 2017].

The limit would always be a unique object that is initial to other objects like it. That is, there exists a unique map from it to other objects like it.

Co-limit is the dual notion.

Definition 3.29 (Finitely Complete Category)**.** *A category is finitely complete if it has equalizers and all finite products. Dually, a category is called co-complete if it has all coequalizers and all finite coproducts.*

Lemma 3.30 (Characterization of finitely complete categories, [Riehl, 2017])**.** *The following conditions are equivalent*

- *$\mathcal{C}$ is finitely complete.*
- *$\mathcal{C}$ has all pullbacks and a terminal object.*

- *$\mathcal{C}$ has all equalizers, binary products, and a terminal object.*

Dually for finitely co-complete categories.

Example 3.31 (Heyting Algebras). *A Heyting algebra is a category that is both finitely complete and finitely cocomplete.*

3.3 Functors and Natural Transformations

Definition 3.32 (Covariant Functor). *Given two categories $\mathcal{C}$ and $\mathcal{D}$, a functor $F : \mathcal{C} \to \mathcal{D}$ is a pair of maps $F_0 : \mathcal{C}_0 \to \mathcal{D}_0$ and $F_1 : \mathcal{C}_1 \to \mathcal{D}_1$ such that*

- *Any arrow $f : A \to B$ in $\mathcal{C}$ is mapped to an arrow $F_1(f) : F_0(A) \to F_0(B)$ in $\mathcal{D}$*
- *For any $A \in \mathcal{C}_0$, $F_0(\mathbf{1}_A) = \mathbf{1}_{F_0(A)}$*
- *For any $f : A \to B$, $F_1(f) : F_0(A) \to F_0(B)$ and for any $f : A \to B$ and $g : B \to C$, $F_1(g \circ f) = F_1(g) \circ F_1(f)$*

Definition 3.33 (Contravariant Functor). *Given categories $\mathcal{C}$, we call a functor contravariant if its domain is the dual of $\mathcal{C}$. That is $F : \mathcal{C}^{op} \to \mathcal{D}$ and for any f, g in $\mathcal{C}$, $F(g \circ f) = F(f) \circ F(g)$.*

Example 3.34 (Morphisms between Heyting Algebras). *Given two Heyting algebras A and B, seen as categories, a functor between these two algebras is a function $f : A \to B$ that respects the order of the two categories. That is $p \leq q$ in A implies that $fp \leq fq$ in B. We call such maps monotonic.*

Definition 3.35 (Category of categories). *The category of categories* **Cat** *is a category such that*

- *The objects of* **Cat** *are the categories.*
- *The morphisms of* **Cat** *are the functors.*

An important example of a category of categories is the category of Heyting algebras treated as categories $\mathbf{Heyt}_{\leq}$ where the objects are Heyting algebras and the maps are the monotone maps. Note that monotone maps do not preserve additional structure of Heyting algebras. That is, given a functor $f : A \to B$, the fact that $a \to b$ in A does not imply that $f(a \to b)$ is $fa \to fb$ in B. When a property of a collection of objects S in A implies a similar property in the image of those objects in B, we say that functor f preserves that structure.

Definition 3.36 (Heyting morphism). *We call a functor f between two Heyting algebras a Heyting morphism if f preserves binary products, binary coproducts, meets, joins and exponential object. That is*

$$f(a \wedge b) = f(a) \wedge f(b)$$
$$f(a \vee b) = f(a) \vee f(b)$$
$$f(a \to b) = f(a) \to f(b)$$
$$f(\bot) = \bot$$
$$f(\top) = \top$$

Likewise, a lattice morphism is a morphism that preserves all binary products and binary coproducts. A meet semi-lattice morphism preserves only the product, etc.

Definition 3.37 (Category of Heyting algebras and category of categories of Heyting Algebras)**.** *The category of Heyting algebras* **Heyt** *is the category with Heyting algebras as objects and Heyting morphisms as arrows. The category of categories of Heyting algebras* $\mathbf{Heyt}_{\leq}$ *is the category with Heyting algebras as objects and monotone maps as arrows.*

Likewise we can think of the category of meet semi-lattices with meet semi-lattice morphisms MeetSemiLat and the category of meet semi-lattice as a category of categories $\text{MeetSemiLat}_{\leq}$. We can do this for all the lattice structures.

We can also talk about arrows between functors

Definition 3.38 (Natural transformation)**.** *Given two functors $F, G : \mathcal{C} \to \mathcal{D}$, a natural transformation $\eta : F \to G$ is a family of arrows $\{\eta_A : F(A) \to G(A)\}_{A \in \mathcal{C}_0} \subseteq \mathcal{D}_1$ such that for any arrow $f : A \to B$ in $\mathcal{C}$, the following diagram commutes:*

$$\begin{array}{ccc} F(A) & \xrightarrow{\eta_A} & G(A) \\ \downarrow{\scriptstyle F(f)} & & \downarrow{\scriptstyle G(f)} \\ F(B) & \xrightarrow{\eta_B} & G(B) \end{array}$$

Definition 3.39 (Natural Isomorphism)**.** *A natural isomorphism is a natural transformation η such that for any object A we have that η_A is an isomorphism.*

3.4 Adjunction

Definition 3.40 (Adjoint functors)**.** *Given two categories $\mathcal{C}$ and $\mathcal{D}$, two functors $L : \mathcal{C} \to \mathcal{D}$ and $R : \mathcal{D} \to \mathcal{C}$ are called adjoint with L left adjoint to R and R right adjoint to L if there exists natural transformations $\eta : \mathbf{1}_{\mathcal{C}} \to RL$ and $\epsilon : LR \to \mathbf{1}_{\mathcal{D}}$ such that $(\varepsilon L) \circ (L\eta) = Id_L$ and $(R\varepsilon) \circ (\eta R) = Id_R$. We say that η is the unit of the adjunction and ϵ is the counit of the adjunction. We denote the adjunction by $L \dashv R$.*

Theorem 3.41 (Adjunctions preserve limits and colimits, [Riehl, 2017])**.** *Left adjoints preserve colimits and right adjoints preserve limits.*

Theorem 3.42 (Characterization of Adjunction, [Riehl, 2017])**.** *Given functors $L : \mathcal{C} \to \mathcal{D}$ and $R : \mathcal{D} \to \mathcal{C}$, the following are equivalent:*

1. *$L \dashv R$*

2. *For any $A \in \mathcal{C}_0$ and $B \in \mathcal{D}_0$, there exists a bijection*

$$\mathcal{D}(LA, B) \cong \mathcal{C}(A, RB)$$

such that for every $A_1, A_2 \in \mathcal{C}$ *and* $B_1, B_2 \in \mathcal{D}$ *and* $f : A_2 \to A_1$, $g : B_1 \to B_2$, *we have that* $\mu_B \circ F(f) = G(f) \circ \mu_A$. *When this condition is met, we say that the following diagram commutes:*

$$\begin{array}{ccc} \mathcal{D}(LA_1, B_1) & \longrightarrow & \mathcal{C}(A_1, RB_1) \\ \Big\downarrow {\scriptstyle f\circ -} & & \Big\downarrow {\scriptstyle -\circ g} \\ \mathcal{D}(LA_2, B_2) & \longrightarrow & \mathcal{C}(A_2, RB_2) \end{array}$$

This criteria is called a natural bijection.

Lemma 3.43 (Uniqueness of Adjunction, [Riehl, 2017])**.** *The left adjoint or right adjoint to a functor, if it exists, is unique up to natural isomorphism.*

3.5 Hyperdoctrines

The last section would explain how to use the category theory to interpret first-order logic. We begin by introducing the language of first-order logic, then develop the categories used for interpreting first-order logic and finish by explaining how this interpetation is done.

3.5.1 Types and Terms

Definition 3.44 (Language)**.** *A language* L *is a triple* $(\mathcal{T}, \mathcal{R}, \mathcal{F})$ *where* $\mathcal{T}, \mathcal{R}, \mathcal{F}$ *are pairwise disjoint sets of type, relation and function symbols respectively, each function symbol* $f \in \mathcal{F}$ *is assigned an arity* $(\sigma_1, \ldots, \sigma_n) \to \tau$ *where* $\sigma_1, \ldots, \sigma_n, \tau$ *are all type symbols and each relation symbol* $R \in \mathcal{R}$ *is assigned an arity* $(\sigma_1, \ldots, \sigma_n)$ *where* $\sigma_1, \ldots, \sigma_n$ *are all type symbols.*

Definition 3.45 (Context)**.** *Given a language* L*, a context in the language* L *is a finite sequence* $\Gamma := (x_1 : \sigma_1, \ldots, x_n : \sigma_n)$ *where* $x_1, \ldots, x_n$ *are pairwise distinct variables and the* σ_i *are types in the language* L*.*

Definition 3.46 (Term)**.** *Let* L *be a language. The terms in the language* L *of type* σ *in context* Γ *are defined by induction as follows:*

1. *If* $x : \sigma \in \Gamma$*, then* x *is a term of type* σ *in context* Γ*.*
2. *If* $t_1, \ldots, t_n$ *are terms of types* $\sigma_1, \ldots, \sigma_n$ *respectively in a context* Γ*, and* f *is a function symbol in* L *of arity* $(\sigma_1, \ldots, \sigma_n) \to \tau$*, then* $f(t_1, \ldots, t_n)$ *is a term of type* τ *in context* Γ*.*

Remark 3.47 (Constants)**.** *Note that this setting also allows us to form constant terms as well. A function symbol from an empty sequence of types* $c : () \to \tau$ *represents a fixed term in the type* τ*.*

We shall show how to interpret terms and contexts in categorical language.

Definition 3.48 (Context morphism). *Let Γ and Δ be contexts in language L with $\Delta = (x_1 : \sigma_1, \ldots, x_n : \sigma_n)$. Then a context morphism $\Gamma \to \Delta$ is a sequence $\vec{s} = (s_1, \ldots, s_n)$ of terms of types $\sigma_1, \ldots, \sigma_n$ respectively, in context Γ.*

Remark 3.49 (The empty context is the terminal object). *Notice that by this definition of context morphism, every context Γ has a unique vacuous mapping to the empty context $()$.*

Definition 3.50 (Term substitution). *Given a term t of type σ in context Δ and a context morphism $\vec{s} = (s_1, \ldots, s_n) : \Gamma \to \Delta$, then the substitution $t[\vec{s}]$ is a term of type σ in context Γ, defined recursively by:*

1. *If $t = x_i$ for $x_i : \sigma_i \in \Delta$, then $t[\vec{s}] = s_i$.*
2. *If $t = f(t_1, \ldots, t_n)$, then $t[\vec{s}] = f(t_1[\vec{s}], \ldots, t_n[\vec{s}])$.*

Notation 3.51 (Constants). *Given a constant $c : () \to \sigma$, we denote the fixed term by $c = c()$.*

A logic above a language consists of the allowed ways of recursively constructing formulas in a given context Γ over language L. We consider for now the most basic construction.

Definition 3.52 (Formula in context). *Let L be a language. A first order formula φ in context Γ in language L is defined recursively by the following grammar:*

$$\top \mid \bot \mid s =_\sigma t \mid R(t_1, \ldots, t_n) \mid \psi \wedge \chi \mid \psi \vee \chi \mid \psi \to \chi \mid \exists x : \sigma.\psi \mid \forall x : \sigma.\psi$$

Where ψ, χ are previously constructed formulas, $s, t, t_1, \ldots, t_n$ are terms in context Γ and R is a relation symbol.

Definition 3.53 (Substitution). *Let φ be a formula in context Δ and $\vec{s} = (s_1, \ldots, s_n) : \Gamma \to \Delta$ be a context morphism. Then the substitution $\varphi[\vec{s}]$ is the formula in context Γ is defined recursively as follows:*

1. *If $\varphi = \top$ or $\varphi = \bot$, then $\varphi[\vec{s}] = \varphi$*
2. *if $\varphi = s =_\sigma t$, then $\varphi[\vec{s}] = s[\vec{s}] =_\sigma t[\vec{s}]$.*
3. *If $\varphi = \chi \Box \psi$, then $\varphi[\vec{s}] = \chi[\vec{s}] \Box \psi[\vec{s}]$ whenever $\Box \in \{\wedge, \vee, \to\}$.*
4. *If $\varphi = R(t_1, \ldots, t_n)$, then $\varphi[\vec{s}] = R(t_1[\vec{s}], \ldots, t_n[\vec{s}])$.*
5. *If $\varphi = Qx : \sigma.\psi$, then $\varphi[\vec{s}] = Qy : \sigma.\psi[\vec{s}, y]$ where $[\vec{s}, y] : \Gamma, y : \sigma \to \Delta, x : \sigma$ is a context morphism (hence $y : \sigma \notin \Gamma$) and $Q \in \{\forall, \exists\}$.*

3.5.2 Hyperdoctrines

Given a context Γ, we can think of the set of formulas in context Γ as a poset $\mathbf{P}\Gamma$. Given a context morphism $\vec{s}\colon \Gamma \to \Delta$, we can consider the substitution as a morphism $\circ[\vec{s}] := \mathbf{P}\vec{s}\colon \Delta \to \Gamma$.

Definitions 3.52 and 3.53 should hint that the image of the mapping $\mathbf{P}$ needs to be a Heyting Algebra with extra-structure that respects the relation symbols and the quantifiers.

Given a projection $\pi : (\Gamma, x : \sigma) \to \Gamma$ and two formulas φ in $\Gamma, x : \sigma$ and ψ in Γ, we have that $\varphi \vdash_{\Gamma, x:\sigma} \psi$, if and only if $\exists x : \sigma.\varphi \vdash_\Gamma \psi$. Dually, for φ in Γ and ψ in $\Gamma, x : \sigma$, we have that $\varphi \vdash_{\Gamma, x:\sigma} \psi$, if and only if $\varphi \vdash_\Gamma \forall x : \sigma.\psi$.

For these conditions to truly represent the existential and universal quantifiers, they need to also satisfy further conditions that would make them behave correctly under substitutions.

Definition 3.54 (Frobenius Condition)**.** *Given a functor in* $\mathbf{Heyt}_{\leq}$, $\pi^{-1} : A \to B$, *with a left adjoint* $\exists_\pi : B \to A$, *we say that* $\exists_\pi$ *satisfies the Frobenius condition if for any* $a \in A$ *and* $b \in B$,

$$\exists_\pi(a \wedge \pi^{-1}(b)) \cong \exists_\pi(a) \wedge b$$

Definition 3.55 (Beck-Chevalley Condition)**.** *Given a category* $\mathcal{C}$ *with all finite products and a functor* $\mathbf{P} : \mathcal{C}^{op} \to \mathbf{Heyt}_{\leq}$, *we say that* $\mathbf{P}$ *satisfies the Beck-Chevalley condition if for any pullback diagram*

$$\begin{array}{ccc}
\Gamma \times_{\pi,\sigma} \Delta & \xrightarrow{\pi'} & \Gamma \\
\big\downarrow{\scriptstyle \sigma'} & & \big\downarrow{\scriptstyle \sigma} \\
\Gamma \times \Delta & \xrightarrow{\pi} & \Gamma
\end{array}$$

where π *and* π' *are projections, we have that* $\exists_{\pi'} \circ P\sigma' \cong P\sigma \circ \exists_\pi$ *and* $\forall_{\pi'} \circ P\sigma' \cong P\sigma \circ \forall_\pi$.

Putting these conditions together, we have that

Definition 3.56 (First Order Hyperdoctrine)**.** *Given a category* $\mathcal{C}$ *with all finite products, a hyperdoctrine over* $\mathcal{C}$ *is a functor* $P : \mathcal{C}^{op} \to \mathbf{Heyt}$ *such that*

1. *For every projection* $\pi : \Gamma \times \Lambda \to \Gamma$, *the map* $P\pi : P\Gamma \to P(\Gamma \times \Lambda)$ *has a left* $\exists_\pi$ *and right* $\forall_\pi$ *adjoints in* $\mathbf{Heyt}_{\leq}$.
2. *For every objects* Γ, Δ, Θ *and pair of maps* $f : \Delta \to \Theta$ *and* $\pi_\Theta : \Gamma \times \Theta \to \Theta$, *the pullback diagram*

$$\begin{array}{ccc}
\Gamma \times \Delta & \xrightarrow{\pi_\Delta} & \Delta \\
\big\downarrow{\scriptstyle (\mathrm{Id}_\Gamma, f)} & & \big\downarrow{\scriptstyle f} \\
\Gamma \times \Theta & \xrightarrow{\pi_\Theta} & \Theta
\end{array}$$

satisfies the Beck-Chevalley condition.

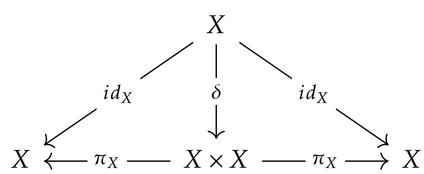
3. *Every projection π satisfies the Frobenius condition.*

We denote $P\pi$ by π^{-1} and call $\mathcal{C}$ the category of contexts.

In general, a hyperdoctrine is a functor from a category of contexts to some poset category (lattice, semi-lattice, etc.) such that some collection of morphisms preserve the Beck-Chevalley and the Frobenius condition.

Lastly, if φ is a formula in context $x:\sigma, y:\sigma$, then we have that $\top \vdash_{x:\sigma} \varphi \Leftrightarrow x =_\sigma y \vdash_{x:\sigma,y:\sigma} \varphi$, This is captured by considering $\exists$ and $\forall$ as respectively the left and right adjoints of the projection map.

Definition 3.57 (Diagonal Map). *Let $\mathcal{C}$ be a category with binary products. For every object X, we have by the definition of the product a unique map $X \to X \times X$ such that the following diagram commutes*



δ is called the diagonal.

Example 3.58 (Diagonal in **Set**). *In* **Set** *the diagonal map is $x \to (x,x)$. Notice that in* **Set** *this map corresponds to the equality relation $= \subseteq X \times X$.*

3.5.3 Interpretations and Soundness

We showed how to represent a first-order theory using types, terms and contexts as well as defined the categories we shall use to interpret formulas over them. We finish by showing how to connect between formulas and categories using interpretations.

Definition 3.59 (Interpretation). *Let* **P** *be a hyperdoctrine with base $\mathcal{C}$ and L be a language. An interpretation of L in the hyperdoctrine* **P** *is a mapping $[\![\cdot]\!]$: which:*

- *Assigns to each type σ in L an object $[\![\sigma]\!]$ in $\mathcal{C}$.*
- *Assigns to each function symbol f with arity $(\sigma_1,\ldots,\sigma_n) \to \tau$ in L a morphism $[\![f]\!] : [\![\sigma_1]\!] \times \cdots \times [\![\sigma_n]\!] \to [\![\tau]\!]$ in $\mathcal{C}$.*
- *Assigns to each relation symbol R with arity $(\sigma_1,\ldots,\sigma_n)$ an element $[\![R]\!] \in$ $\mathbf{P}([\![\sigma_1]\!] \times \cdots \times [\![\sigma_n]\!])$.*

Given $\Gamma = x_1 : \sigma_1, \ldots, x_n : \sigma_n$, denote $[\![\Gamma]\!] = [\![\sigma_1]\!] \times \ldots [\![\sigma_n]\!]$.

Definition 3.60 (Interpretation of terms). *If t is a term of type σ in context Γ, then the interpretation $[\![t]\!] : [\![\Gamma]\!] \to [\![\sigma]\!]$ is defined by recursively as:*

1. *For $x_i : \sigma_i \in \Gamma$, we have that the interpretation of $x_i : \sigma_i$ is the i^{th} projection $[\![x_i]\!] = \pi_i : [\![\Gamma]\!] \to [\![\sigma_i]\!]$.*

2. *if $t_1, \ldots, t_n$ are terms of types $\sigma_1, \ldots, \sigma_n$ in context Γ, f is a function symbol with arity $(\sigma_1, \ldots, \sigma_n) \to \tau$ and the interpretation of each t_i is $[\![t_i]\!] : [\![\Gamma]\!] \to [\![\sigma_i]\!]$, then $[\![f(t_1, \ldots, t_n)]\!]$ is the composition: $[\![f]\!] \circ ([\![t_1]\!], \ldots, [\![t_n]\!])$.*

3. *If $\vec{s} = (s_1, \ldots, s_n) : \Gamma \to \Delta$ is a context morphism, we can also define: $[\![\vec{s}]\!] = ([\![s_1]\!], \ldots, [\![s_n]\!]) : [\![\Gamma]\!] \to [\![\Delta]\!]$*

Definition 3.61 (Lawvere Equality)**.** *We define the diagonal map $\delta : \Gamma \times \Gamma \to \Gamma$ as the map $(\mathbf{Id}_\Gamma, \mathbf{Id}_\Gamma)$. The Lawvere equality on Γ is the element $\mathrm{Eq}_\Gamma \in \mathbf{P}(\Gamma \times \Gamma)$ given by:*

$$\mathrm{Eq}_\Gamma = \exists_\delta(\top)$$

Definition 3.62 (Interpretation of formulas)**.** *Given a formula φ in context Γ, the interpretation of φ is defined recursively on Γ as:*

1. *$[\![\top]\!]$ and $[\![\bot]\!]$ are the $\top$ and $\bot$ elements of $\mathbf{P}\Gamma$.*

2. *Given terms s, t if type σ in context Γ and the interpretation $([\![s]\!], [\![t]\!]) : [\![\Gamma]\!] \to [\![\sigma]\!] \times [\![\sigma]\!]$, then $[\![s =_\sigma t]\!]$ is $([\![s]\!], [\![t]\!])^{-1}(\mathrm{Eq}_{[\![\sigma]\!]})$.*

3. *Given terms $t_1, \ldots, t_n$ of types $\sigma_1, \ldots, \sigma_n$ in context Γ, a relation symbol R of arity $(\sigma_1 \ldots, \sigma_n)$ and the interpretation $([\![t_1]\!], \ldots, [\![t_n]\!]) : [\![\Gamma]\!] \to [\![\sigma_1]\!] \times \cdots \times [\![\sigma_n]\!]$, then $[\![R(t_1, \ldots t_n)]\!]$ is $([\![t_1]\!], \ldots, [\![t_n]\!])^{-1}([\![R]\!])$.*

4. *If φ and ψ are formulas in context Γ, then $[\![\varphi \Box \psi]\!]$ is $[\![\varphi]\!] \Box [\![\psi]\!]$ for $\Box \in \{\wedge, \vee, \to\}$.*

5. *If φ is a formula in context $\Gamma, x : \sigma$ and $\pi : [\![\Gamma]\!] \times [\![x : \sigma]\!] \to]\!]\Gamma]\!]$ then $[\![Qx : \sigma . \varphi]\!]$ is $Q_\pi([\![\varphi]\!])$ for $Q \in \{\exists, \forall\}$.*

Definition 3.63 (Validity)**.** *Given formulas φ and ψ in context Γ, we say that $\varphi \vdash \psi$ is valid under the interpretation $[\![\cdot]\!]$ on hyperdoctrine $\mathbf{P}$ if $[\![\varphi]\!] \leq [\![\psi]\!]$ in $\mathbf{P}\Gamma$. We say that φ is valid if $\top \vdash \varphi$ is valid.*

We say that $\varphi \vdash \psi$ is true under an interpretation $[\![\varphi]\!] \leq [\![\psi]\!]$ if it is valid under all contexts.

Definition 3.64 (Soundness)**.** *Given an interpretation $[\![\cdot]\!]$ on a hyperdoctrine $\mathbf{P}$ and a logic Λ, we say that Λ is sound with respect to $[\![\cdot]\!]$ if $\varphi \vdash_\Lambda \psi$ entails that $\varphi \vdash \psi$ is true under $[\![\cdot]\!]$.*

Example 3.65 (The hyperdoctrine of **Set**)**.** *Consider the functor that maps each set to the lattice of its powerset*

$$\begin{aligned} \mathbf{P} : \mathbf{Set} &\to \mathrm{Latt} \\ A &\mapsto (\mathcal{P}A, \subseteq, \cap, \cup, \to, A, \varnothing) \end{aligned}$$

where $U \to V = U^C \cup V$ for any $U, V \subseteq A$.

For any projection map $\pi : A \times B \to A$ that

$$\begin{aligned}
\pi^{-1} &: S \mapsto \{(a,b) \in A \times B \mid a \in S\} \\
\exists_\pi &: S \mapsto \{a \in A \mid \textit{there exists } b \in B \textit{ such that } (a,b) \in S\} \\
\forall_\pi &: S \mapsto \{a \in A \mid \textit{for all } a \in A \textit{ we have that } (a,b) \in S\}
\end{aligned}$$

In the hyperdoctrine of sets, every function $f : A \to B$ has left and right adjoints:

$$\begin{aligned}
f^{-1} &: S \mapsto \{a \in A \mid f(a) \in S\} \\
\exists_f &: S \mapsto \{b \in B \mid \textit{there exists } a \in A \textit{ such that } f(a) = b \textit{ and } a \in S\} \\
\forall_f &: S \mapsto \{b \in B \mid \textit{for all } a \in A \textit{ such that } f(a) = b \textit{ we have that } a \in S\}
\end{aligned}$$

The empty context is the singleton set $\{*\}$. Note that indeed every set has a unique map to the singleton set. Furthermore, each map $s : \{*\} \to A$ fixes an element s in A.

$$s^{-1} : S \mapsto \begin{cases} \{*\} & s \in S \\ \varnothing & s \notin S \end{cases}$$

Part II

Box Spaces

Chapter 4

Box Spaces

In this section we use categorical logic to formalize the interpretation of $\mathcal{EL}^{++}$ using boxes found in [Jackermeier et al., 2024b]. [Jackermeier et al., 2024b] suggests that their interpretation adheres to the underlying semantics of $\mathcal{EL}^{++}$ while offering a computationally efficient embedding of the knowledge due to the usage of computationally nice objects (boxes). We will use the semantics found in 3.5.1 to formalize the box system. We will finish this chapter by showing that this system is not complete with respect to $\mathcal{EL}^{++}$.

4.1 The $\mathcal{EL}^{++}$ Language

Definition 4.1 ($\mathcal{EL}^{++}$ language)**.** *Given a signature $(N_\mathcal{C}, \mathcal{R}, \mathcal{O})$, as defined in definition (1.1), the corresponding $\mathcal{EL}^{++}$ language is $(\{\sigma\}, \mathcal{R} \cup N_\mathcal{C} \cup \mathcal{O}, \varnothing)$.*

Where the arity of any relation symbol $R \in \mathcal{R}$ is (σ, σ), the arity of any concept symbol $C \in N_\mathcal{C}$ and the arity of any individual $a \in \mathcal{O}$ is σ.

Definition 4.2 ($\mathcal{EL}^{++}$ formula)**.** *An $\mathcal{EL}^{++}$ formula φ in context Γ is defined recursively by the following grammar:*

$$\top \mid \bot \mid R(t_1, t_2) \mid C(t) \mid \psi \wedge \chi \mid \exists x : \sigma . R(x, t) \wedge \varphi(t)$$

where φ, ψ are previously constructed formulas, t, t_1, t_2 are terms in context Γ, a is a constant symbol, R is a relation symbol and C is a concept symbol.

Notice that the $\mathcal{EL}^{++}$ concept $\exists R.\varphi$ is interpreted as $\exists x : \sigma . R(x, t) \wedge \varphi(t)$.

The rules for term and formula substitution are as found in definitions 3.50, 3.53.

Lastly, notice that we do not fix terms. That is, maps $() \to \sigma$ do not exist in our language. The reason we choose to represent individuals as boxes is that machine learning algorithms struggle differentiating between very small boxes and boxes with volume zero. Hence, it is preferable to represent individuals as boxes with non-zero volume. If we were to use maps $() \to \sigma$ to represent individuals, then they would be represented as singleton elements in our space.

Definition 4.3 ($\mathcal{EL}^{++}$ Hyperdoctrine)**.** *Given a category $\mathcal{C}$ with all finite products, a $\mathcal{EL}^{++}$ hyperdoctrine over $\mathcal{C}$ is a functor $P : \mathcal{C}^{op} \to$ BoundMeetSemiLat where* BoundMeetSemiLat *is the category of bounded meet semi-lattices, such that*

1. *For every projection $\pi : \Gamma \times \Lambda \to \Gamma$, the map $P\pi : P\Gamma \to P(\Gamma \times \Lambda)$ has a left adjoint $\exists_\pi$ in* $\text{BoundMeetSemiLat}_{\leq}$.
2. *$\exists_\pi$ satisfies the Beck-Chevalley condition and the Frobenius reciprocity.*

4.2 Box Space Category

We first construct the box space category.

Definition 4.4 (Box)**.** *Given a subset S of $\mathbb{R}^n$, we define a box over S as a subset of the form $\{\vec{x} \in S : m_i \leq x_i \leq M_i\}$ where for any i, we have that $m_i, M_i \in \mathbb{R} \cup \{-\infty, \infty\}$*

We denote a box by $B_{\mathcal{B}}(\vec{m}, \vec{M})$ or B when we do not care about the precise coordinates and underlying subset. We denote $\infty = (\infty, \ldots, \infty)$ and $-\infty = (-\infty, \ldots, -\infty)$. Note that $B(-\infty, \infty) = S$ and $B(-\infty, -\infty) = B(\infty, \infty) = \varnothing$.

Definition 4.5 (Box space)**.** *Given $S \subseteq \mathbb{R}^n$, we define a box space $\mathcal{B}$ over S as a set of boxes of S where*

- *$S, \varnothing \in \mathcal{B}$.*
- *$\mathcal{B}$ is closed under arbitrary intersections.*

Example 4.6 (Discrete box space)**.** *For any subset S of $\mathbb{R}^n$, we have that $(S, \{B \subseteq S \mid B$ is a box in $S\})$ is a box space.*

Notice that $\mathcal{B}$ induces a meet semi-lattice $(\mathcal{B}, \sqsubseteq, \sqcap)$.

Definition 4.7 (Box space morphism)**.** *Given two box spaces $(S, \mathcal{B})$ and $(S', \mathcal{B}')$, we define a box space morphism $f : (S, \mathcal{B}) \to (S', \mathcal{B}')$ as a function $f : S \to S'$ such that for all $B' \in \mathcal{B}'$ we have that $f^{-1}[B'] \in \mathcal{B}$.*

Since preimages preserve intersections, we have that

Lemma 4.8. *Every box space morphism $f : (S, \mathcal{B}) \to (S', \mathcal{B}')$ induces a meet semi-lattice morphism $f^{-1} : \mathcal{B}' \to \mathcal{B}$.*

Lemma 4.9. *For any two box space morphisms $f : (S, \mathcal{B}) \to (S', \mathcal{B}')$ and $g : (S', \mathcal{B}') \to (S'', \mathcal{B}'')$, we have that $g \circ f$ is a box space morphism.*

These results lead to a category **Box** with box spaces as objects and box space morphisms as arrows.

Furthermore, we have a functor from the box category to the category of meet semi-lattices $U : \mathbf{Box}^{op} \to$ MeetSemiLat:

$$U_0 : (S, \mathcal{B}) \mapsto \mathcal{B}$$
$$U_1 : f(\cdot) \mapsto f^{-1}[\cdot]$$

We shall show that this induces a hyperdoctrine.

Category of contexts

We begin by showing that **Box** contains all finite products. We shall show this by proving that **Box** has all binary products and a terminal object. We define the concatenation of two vectors $\vec{x} := (x_1, \ldots, x_n), \vec{y} := (y_1, \ldots, y_m)$ as $(\vec{x}, \vec{y}) := (x_1, \ldots, x_n, y_1, \ldots, y_m)$.

Denote the product of two boxes:

$$B(\vec{m}, \vec{M}) \times B(\vec{m}', \vec{M}') = B((\vec{m}, \vec{m}'), (\vec{M}, \vec{M}'))$$

Definition 4.10 (Product box space)**.** *Given two box spaces $(S, \mathcal{B})$ and $(S', \mathcal{B}')$, we define their product box space $(S \times S', \mathcal{B} \times \mathcal{B}')$ as the set of all subsets of the form $B \times B'$ with $B \in \mathcal{B}$ and $B' \in \mathcal{B}'$.*

Lemma 4.11. *The product box space is the product in* **Box***.*

Proof. As every box morphism is also a function of the underlying sets, it suffices to show that the projections $\pi_S : S \times S' \to S, \pi'_{S'} : S \times S' \to S'$ are box morphisms and that for every box space $(C, \mathcal{C})$ with box morphisms $f : (C, \mathcal{C}) \to (S, \mathcal{B})$, $g : (C, \mathcal{C}) \to (S', \mathcal{B}')$ the unique morphism $(f, g) : c \to (f(c), g(c))$ is a box morphism.

Given box spaces $(S, \mathcal{B})$, $(S', \mathcal{B}')$ and a box $B(\vec{m}, \vec{M}) \in \mathcal{B}$, notice that $\pi^{-1}[B(\vec{m}, \vec{M})] = B(\vec{m}, \vec{M}) \times B(-\infty, \infty)$. As this is inside of $\mathcal{B} \times \mathcal{B}'$, we get that for every box in $\mathcal{B}$, its preimage is in $\mathcal{B} \times \mathcal{B}'$.

Given a box space $(A, \mathcal{S})$ and two box morphisms $f : (A, \mathcal{S}) \to (S, \mathcal{B})$ and $g : (A, \mathcal{S}) \to (S', \mathcal{B}')$, the unique map of sets $(f, g) : x \to (f(x), g(x))$ that commutes with the projections is also a box morphism. This is because for any $B \times B' \in \mathcal{B} \times \mathcal{B}'$, we have that

$$(f, g)^{-1}[B \times B'] = f^{-1}[B] \cap g^{-1}[B']$$

As f and g are box morphisms, the preimages are in the box space $\mathcal{S}$. As box spaces are closed under intersection, $f^{-1}[B] \cap g^{-1}[B']$ is in the box space. Thus, (f, g) is a box morphism. □

Notice that every pair of box spaces has a binary product.

Lemma 4.12 (The terminal object in **Box**)**.** *The box space $(\{0\}, \{\emptyset, \{0\}\})$ is the terminal object in* **Box***.*

Proof. Given any box space $(A, \mathcal{B})$, there exists a single unique map $! : A \to \{0\}$. This map is a box morphism because $!^{-1}[\{0\}] = A \in \mathcal{B}$. □

By Lemma Theorem 3.20, these results imply that

Lemma 4.13. **Box** *has all finite products.*

Left adjoint

To define the existential quantifier, we first define a closure operator.

Definition 4.14 (Closure operator)**.** *Given a set S, a closure operator is a function* $\mathrm{Cl} : \mathcal{P}S \to \mathcal{P}S$ *that satisfies the following conditions*

1. $A \subseteq \mathrm{Cl}(A)$ *(extensivity).*
2. $A \subseteq B$ *implies* $\mathrm{Cl}(A) \subseteq \mathrm{Cl}(B)$ *(monotinicity).*
3. $\mathrm{Cl}(\mathrm{Cl}(A)) \subseteq \mathrm{Cl}(A)$ *(idempotency).*

Definition 4.15 (Box closure operator)**.** *Given a box space $(S, \mathcal{B})$, we define the box closure operator*

$$\begin{aligned} \mathrm{Cl} : \mathcal{P}(S) &\to \mathcal{B} \\ A &\mapsto \bigcap \{B \in \mathcal{B} : A \subseteq B\} \end{aligned}$$

Remark 4.16. *Notice that the box closure operator is the left adjoint to the forgetful functor*

$$\begin{aligned} U : \mathcal{B} &\to \mathcal{P}(S) \\ B &\mapsto B \end{aligned}$$

Given a subset $K \subseteq S$ and a box $B \in \mathcal{B}$, we have that $\mathrm{Cl}(K) \subseteq B$ *if and only if* $K \subseteq \mathrm{Cl}(B) = B$.

We notice that the closure operator satisfies the closure conditions: for any $A \subseteq S$, we have that

1. $A \subseteq \mathrm{Cl}(A)$ (extensivity).
2. $A \subseteq B$ implies $\mathrm{Cl}(A) \subseteq \mathrm{Cl}(B)$ (isotonicity).
3. $\mathrm{Cl}(\mathrm{Cl}(A)) \subseteq \mathrm{Cl}(A)$ (idempotency).

Furthermore, for any box $B \in \mathcal{B}$, we have that $\mathrm{Cl}(B) = B$.

Definition 4.17 (Existential quantifier in **Box**)**.** *We define the existential quantifier as:*

$$\begin{aligned} \exists_{(\cdot)} : \mathbf{Box}((S, \mathcal{B}), (S', \mathcal{B}')) &\to \mathbf{Cat}(\mathcal{B}, \mathcal{B}') \\ f(\cdot) &\mapsto \mathrm{Cl} \circ f[\cdot] \end{aligned}$$

It is easy to see that the existential quantifier when restricted to boxes is a functor between two meet semi-lattices as $\mathrm{Cl} \circ f$ is always monotone.

Remark 4.18. *Notice that for any projection π we have that the existential quantifier restricted to boxes is $\exists_\pi[\circ] = \pi[\circ]$. This fact is important because the later categories we would construct would not have a general existential quantifier but only a quantifier for projection when restricted to the objects of the underlying poset.*

Lemma 4.19 (Existential quantifier is a left adjoint)**.** *For any morphism*

$$\pi : (S, \mathcal{B}) \to (S', \mathcal{B}')$$

the existential operator $\exists_\pi$ is the left adjoint of π^{-1} in MeetSemiLat$_{\leq}$*.*

Proof. We shall show that for any $B \in \mathcal{B}$ and $B' \in \mathcal{B}'$,

$$\exists_\pi[B] \subseteq B' \text{ if and only if } B \subseteq \pi^{-1}[B']$$

$\Rightarrow$ Suppose that $Cl \circ \pi[B] = \exists_\pi[B] \subseteq B'$. Let $\vec{x} \in B$. Then by extensivity of closure $\pi(\vec{x}) \in \pi[B] \subseteq Cl \circ \pi[B]$. Hence $\pi(\vec{x}) \in B'$. Thus $\vec{x} \in \pi^{-1}[B']$.

$\Leftarrow$ Suppose that $B \subseteq \pi^{-1}[B']$. Then $\pi[B] \subseteq B'$. By isotonicity $Cl(\pi[B]) \subseteq Cl(B') = B'$. □

Notice that the proof of the left adjunction relies solely on the closure properties.

By Theorem 3.42, we also get the following property

Corollary 4.20. *For any morphism $\pi : (S, \mathcal{B}) \to (S', \mathcal{B}')$ and objects $B \in \mathcal{B}$ and $B' \in \mathcal{B}'$, we have that $\exists_\pi \circ \pi^{-1} B \subseteq B$ and $B' \subseteq \pi^{-1} \circ \exists_\pi B'$*

Remark 4.21 (Existential quantifier and the product box space)**.** *Given a product box space and a projection $\pi : (S \times S', \mathcal{B} \times \mathcal{B}') \to (S, \mathcal{B})$, we have in particular that*

$$\exists_\pi(B \times B') = \{\vec{x} \in S \mid \exists \vec{x'} \in \mathcal{B}' : (\vec{x}, \vec{x'}) \in B \times B'\} \in \mathcal{B}$$

We need to show that the existential operator satisfies the Beck-Chevalley condition and the Frobenius reciprocity.

Lemma 4.22 (Beck-Chevalley Condition for **Box**)**.** *Given any three box spaces $(A, \mathcal{A}), (C, \mathcal{C})$ and $(D, \mathcal{D})$, a map $f : (C, \mathcal{C}) \to (D, \mathcal{D})$ and pullback diagram*

$$\begin{array}{ccc}
(A \times C, \mathcal{A} \times \mathcal{C}) & \xrightarrow{\ \pi\ } & (C, \mathcal{C}) \\
\Big\downarrow {\scriptstyle (Id_A, f)} & & \Big\downarrow {\scriptstyle f} \\
(A \times D, \mathcal{A} \times \mathcal{D}) & \xrightarrow{\ \pi'\ } & (D, \mathcal{D})
\end{array}$$

in **Box***, the Beck-Chevalley condition holds in* MeetSemiLat.

$$\exists_\pi \circ (Id_A, f)^{-1} = f^{-1} \circ \exists_{\pi'}$$

Proof. Given $B \times B' \in \mathcal{A} \times \mathcal{D}$, we have that

$$\exists_\pi \circ (Id_A, f)^{-1}(B \times B') = \mathrm{Cl}\, \pi(Id_A, f)^{-1}(B \times B') = \mathrm{Cl}\, \pi(B \times f^{-1}[B']) = \mathrm{Cl}\, f^{-1}[B'] = f^{-1}[B']$$
$$f^{-1} \circ \exists_{\pi'}(B \times B') = f^{-1}\mathrm{Cl}\, \pi'(B \times B') = f^{-1}\mathrm{Cl}(B') = f^{-1}[B']$$

□

To finish the construction of the hyperdoctrine, we need to show that the existential operator satisfies the Frobenius reciprocity.

Lemma 4.23 (Frobenius reciprocity for **Box**). *Given a projection box space morphism $\pi : (A\times B, \mathcal{A}\times\mathcal{B}) \to (A, \mathcal{A})$, the existential operator $\exists_\pi$ satisfies the Forbenius reciprocity. That is, for any two boxes $B\times C \in \mathcal{A}\times\mathcal{B}$ and $B' \in \mathcal{A}$,*

$$\exists_\pi(\pi^{-1}(B')\cap(B\times C)) = B'\cap\exists_\pi(B\times C)$$

Proof. Unpacking the definitions:

$$\exists_\pi(\pi^{-1}(B')\cap(B\times C)) = \mathrm{Cl}\,\pi((B'\times\infty)\cap(B\times C)) = \mathrm{Cl}\,\pi((B'\cap B)\times C) = \mathrm{Cl}(B'\cap B) = B'\cap B$$
$$B'\cap\exists_\pi(B\times C) = B'\cap\mathrm{Cl}\,\pi(B\times C) = B'\cap B$$

□

Combining the two lemmas, we can conclude the following statement:

Theorem 4.24 (Box spaces form a hyperdoctrine). *The functor* **Box** $\to$ BoundMeetSemiLat *is a $\mathcal{EL}^{++}$ hyperdoctrine.*

4.3 Box Interpretations

Our interpretation of $\mathcal{EL}++$ follows directly from definitions 3.60 and 3.62. We assign σ to $S\subseteq\mathbb{R}^n$, each relation symbol R to a box in $\mathcal{B}$ of $S\times S$ and each individual a to a box $\{a\}$ that is atomic in $\mathcal{B}$.

Definition 4.25 (Categorical Semi-Interpretation of $\mathcal{EL}^{++}$ in **Box**). *Given a $\mathcal{EL}^{++}$ language $(\{\sigma\}, \mathcal{R}\cup N_{\mathcal{C}}\cup\mathcal{O}, \varnothing)$, a box space Semi-interpretation of the language is a box space $(S, \mathcal{B})$, along with a mapping*

$$[\![\cdot]\!] : N_{\mathcal{C}} \to \mathcal{B}$$
$$[\![\cdot]\!] : \mathcal{R} \to \mathcal{B}\times\mathcal{B}$$
$$[\![\cdot]\!] : \mathcal{O} \to \mathcal{B}\setminus\{\bot\}$$

Notice that our language does not have any functions. However, the trivial projection $\chi : X \cong \{*\}\times S \to \{*\}$ that sends all of S to $\{*\}$ has an inverse $\chi^{-1} : \{\top,\bot\} := \{\{*\},\varnothing\} \to S$ that sends $\top$ to S and $\bot$ to $\varnothing$. The left adjoint of this projection $\exists_\chi := \chi$ sends non-empty sets to $\top$ and the empty set to $\bot$. This map is important because it allows us to argue within the language that a formula C in context S is not empty: $\top \subseteq \exists_\chi.C$.

Given a formula φ in context σ, the interpretation of φ is defined recursively on σ as:

1. *If φ is $\top$ or $\bot$ then $[\![\varphi]\!]$ is S or $\varnothing$ respectively.*
2. *If $\varphi := \psi\wedge\chi$, then $[\![\varphi]\!] = [\![\psi\wedge\chi]\!] := [\![\varphi]\!]\cap[\![\psi]\!]$.*

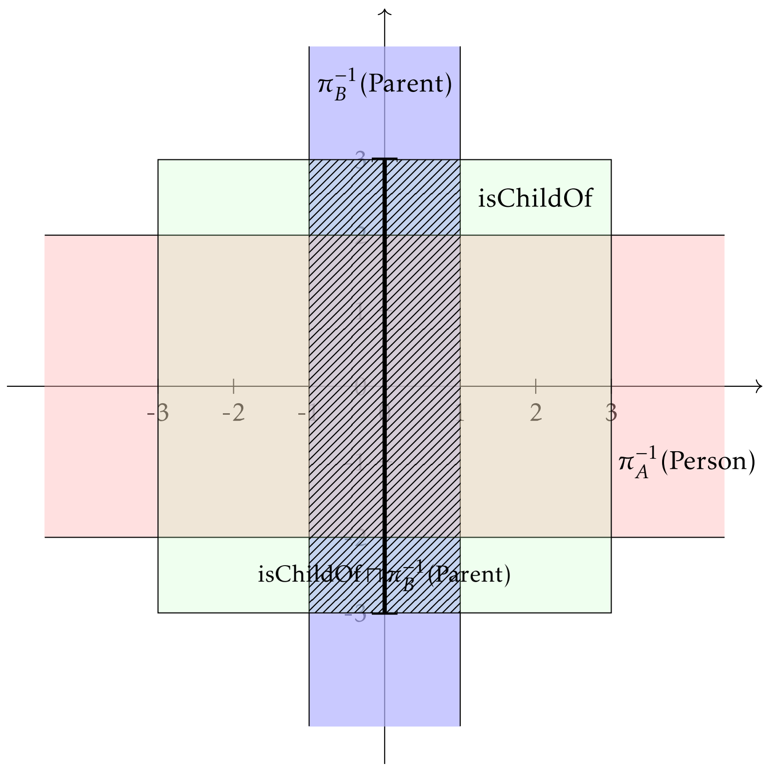


Figure 4.1: Visualization of the box representation of Example 4.28. The bold line is ∃isChildOf.Parent.

3. *If $\varphi := \exists x : \sigma.\psi$ where ψ is in context $x : \sigma, y : \sigma$ with projections π_1 and π_2, then $[\![\varphi]\!] = [\![\exists x : \sigma.\psi]\!] := \mathrm{Cl} \circ \pi_1([\![\psi]\!])$.*

Definition 4.26 (Categorical Interpretation of $\mathcal{EL}^{++}$)**.** *Given a $\mathcal{EL}^{++}$ semi-interpetation $((S, \mathcal{B}), [\![\cdot]\!])$ is a $\mathcal{EL}^{++}$ interpretation if it satisfies the following additional condition*

- *If $[\![\psi]\!] \subseteq [\![\{a\}]\!]$ for some $\psi \neq \bot$, then $[\![\{a\}]\!] \subseteq [\![\psi]\!]$.*

Recall that $\exists R.C$ is read as $\exists y : \sigma.R \wedge C$. That is

$$[\![\exists y : \sigma.R(x,y) \wedge C(y)]\!] = Cl \circ \pi_1([\![R(x,y) \wedge C(y)]\!]) = Cl \circ \pi_1([\![R]\!] \cap \pi_2^{-1}[\![C]\!])$$

Notation 4.27. *We denote $\exists R.C := \exists y : \sigma.R(x,y) \wedge C(y)$ and $\forall R.C := \forall y : \sigma.R(x,y) \wedge C(y)$.*

Case Example

We shall show how to interpret the description logic existential quantifier in our system. [Xiong et al., 2022a] used the family relation dataset to originally motivate the implementation of box space approaches, as they are able to express the complicated operators of $\mathcal{EL}^{++}$ with simple spatial operations.

$$\text{Parent} \sqsubseteq \text{Person} \tag{4.1}$$
$$\text{Person} \sqsubseteq \exists \text{isChildOf.Parent} \tag{4.2}$$

Figure **??** demonstrates how such an embedding would be represented using our box space representation.

Example 4.28. *A model of (4.1) and (4.2) is two box spaces* $A := (\mathbb{R}, \{\mathbb{R}, \text{Parent}, \text{Person}, \varnothing\})$ *and* $B := (\mathbb{R}, \{\mathbb{R}, \text{Parent}, \text{Person}, \varnothing\})$ *such that* $\text{Parent} := [-1,1], \text{Person} := [-2,2]$ *and* $\text{isChildOf} := [-3,3] \times [-3,3]$*. Notice that*

$$\begin{aligned}
\exists \text{isChildOf.Parent} &= \exists_{\pi_A}(\text{isChildOf} \sqcap \pi_B^{-1}(\text{Parent})) \\
&= Cl \circ \pi_A([-3,3] \times [-3,3] \sqcap [-\infty,\infty] \times [-1,1]) \\
&= Cl \circ \pi_A([-3,3] \times [-1,1]) \\
&= Cl([-3,3]) = [-3,3] \Rightarrow \\
\text{Person} = [-1,1] \sqsubseteq [-3,3] &= \exists \text{isChildOf.Parent}
\end{aligned}$$

Thus both conditions are met.

4.3.1 Model Theoretic Results

Every Satisfiable Knowledge Base has a Finite Set Theoretic Model

We will show that every satisfiable knowledge base $\mathcal{K}$ in $\mathcal{EL}^{++}$ has a finite model. Our result follows from [Baader et al., 2017] and we will use results from that book whenever needed.

We will first prove the following result about $\mathcal{EL}^{++}$:

Theorem 4.29. *Given any knowledge base $\mathcal{K}$ of $\mathcal{EL}^{++}$, if $\mathcal{K}$ is satisfiable, then there exists a finite model of $\mathcal{K}$.*

We begin by defining some important concepts:

Definition 4.30 (Size and Subconcepts)**.** *Given a concept C of $\mathcal{EL}^{++}$, we define the set of subconcepts* $\text{sub}(C)$ *and the size* $\text{size}(C)$ *recursively as*

- *If C is a basic concept $A \in \text{BC}$, then* $\text{size}(C) = 1$ *and* $\text{sub}(C) = \{A\}$.
- *If $C = C_1 \sqcap C_2$ then* $\text{size}(C) = 1 + \text{size}(C_1) + \text{size}(C_2)$ *and* $\text{sub}(C) = \{C\} \cup \text{sub}(C_1) \cup \text{sub}(C_2)$.
- *If $C = \exists R.D$, then* $\text{size}(C) = 1 + \text{size}(D)$ *and* $\text{sub}(C) = \{C\} \cup \text{sub}(D)$.

Given an ABox $\mathcal{A}$, we define $\text{sub}(\mathcal{A}) := (\bigcup_{a:C \in \mathcal{A}} \text{sub}(C) \cup \{a\}) \cup (\bigcup_{(a,b):R \in \mathcal{A}} \text{sub}(C) \cup \{a\} \cup \{b\})$ *and* $\text{size}(\mathcal{A}) := \sum_{a:C \in \mathcal{A}} \text{size}(C)$*. Given a TBox $\mathcal{T}$, we define* $\text{sub}(\mathcal{T}) := \bigcup_{C \sqsubseteq D \in \mathcal{T}} \text{sub}(C) \cup \text{sub}(D)$ *and* $\text{size}(\mathcal{A}) := \sum_{C \sqsubseteq \in \mathcal{T}} \text{size}(C) + \text{size}(D)$*. Given a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, we define* $\text{sub}(\mathcal{K}) := \text{sub}(\mathcal{T}) \cup \text{sub}(\mathcal{A})$ *and* $\text{size}(\mathcal{K}) := \text{size}(\mathcal{T}) + \text{size}(\mathcal{A})$.

Let $M := (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a model of a closed knowledge base $\mathcal{K}$. Define the equivalence relation $\sim$ between elements of $\Delta^{\mathcal{I}}$:

$$x \sim y \text{ if and only if for any } C \in \text{sum}(\mathcal{K}) \text{ we have that } x \in C \text{ if and only if } y \in C$$

Denote the equivalence classes generated by $\sim$ as $|x|$.

Definition 4.31 (Filtrated model)**.** *Given a model $M := (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of a knowledge base K, we define a filtrated model $M^f := (\Delta^{\mathcal{I}}_f, \cdot^{\mathcal{I}}_f)$ to be a $\mathcal{EL}^{++}$ model with $\Delta^{\mathcal{I}}_f = \{|x| \mid x \in \Delta^{\mathcal{I}}\}$ and for any basic concept A in* sub$(\mathcal{K})$ *and role $R \in \mathcal{R}$ in* sub$(\mathcal{K})$ *we have that $A^{\mathcal{I}}_f = \{|x| \mid x \in A^{\mathcal{I}}\}, R^{\mathcal{I}}_f = \{(|x|,|y|) \mid \exists x' \in |x|, y' \in |y| : (x',y') \in R^{\mathcal{I}}\}$.*

Notice that for any $x, y \in \Delta^{\mathcal{I}}$ we have that if $(x,y) \in R^{\mathcal{I}}$ then $(|x|,|y|) \in R^{\mathcal{I}}_f$. We prove a useful lemma.

Lemma 4.32. *Given a role R and elements $x, y \in \Delta^{\mathcal{I}}$, if $(|x|,|y|) \in R^{\mathcal{I}}_f$, then for any $\exists R.C \in \text{sub}(\mathcal{K})$, we have that $y \in C^{\mathcal{I}}$ implies $x \in (\exists R.C)^{\mathcal{I}}$.*

Proof. Since $(|x|,|y|) \in R^{\mathcal{I}}_f$, there exists $x' \in |x|, y' \in |y|$ such that $(x',y') \in R^{\mathcal{I}}$. Since $\exists R.C \in \text{sub}(\mathcal{K})$, we also have that $C \in \text{sub}(\mathcal{K})$. As $y \in C^{\mathcal{I}}$ and $y \sim y'$, we have that $y' \in C^{\mathcal{I}}$. This implies that $x' \in (\exists R.C)^{\mathcal{I}}$. Since $x \sim x'$, this means that $x \in (\exists R.C)^{\mathcal{I}}$ □

Lemma 4.33. *Let M^f be a filtrated model of M through knowledge base $\mathcal{K}$. Then for any $C \in \text{sub}(\mathcal{K})$, and all $x \in \Delta^{\mathcal{I}}$, we have that $x \in C^{\mathcal{I}}$ if and only if $|x| \in C^{\mathcal{I}}_f$.*

Proof. We proceed through induction on the size of C. Suppose that C is of size 1. Then it follows from definition that $x \in C^{\mathcal{I}}$ if and only if $|x| \in C^{\mathcal{I}}_f$. For any concept C of size $n > 1$, one of the following cases hold:

- $C = C_1 \sqcap C_2$ for concepts C_1, C_2 of size less than C. We have that $D, L \in \text{sub}(\mathcal{K})$. By induction hypothesis, $x \in C^{\mathcal{I}}_i$ if and only if $|x| \in C^{\mathcal{I}}_i$ for $i \in \{1,2\}$. Which imply that $x \in (C_1 \sqcap C_2)^{\mathcal{I}}$ if and only if $x' \in (C_1 \sqcap C_2)^{\mathcal{I}}_f$.

- $C = \exists R.D$ for D of size less than C. Suppose that $x \in (\exists R.D)^{\mathcal{I}}$. Then there exists $y \in D^{\mathcal{I}}$ such that $(x,y) \in R^{\mathcal{I}}$. Since $D \in \text{sub}(\mathcal{K})$, we have by induction hypothesis that $|y| \in D^{\mathcal{I}}_f$. Since $(|x|,|y|) \in R^{\mathcal{I}}_f$, we have that $|x| \in (\exists R.D)^{\mathcal{I}}_f$.

  Suppose that $|x| \in (\exists R.D)^{\mathcal{I}}_f$. Then there exists $|y| \in D^{\mathcal{I}}_f$ such that $(|x|,|y|) \in R^{\mathcal{I}}_f$. By induction hypothesis, $y \in D^{\mathcal{I}}$. By Lemma 4.32, $x \in (\exists R.D)^{\mathcal{I}}$.

As these are the only cases, this proves the induction hypothesis. □

The above lemma shows that a filtrated model is a model of $\mathcal{K}$.
Consider the mapping

$$\Delta^{\mathcal{I}}_f \to \mathcal{P}(\text{sub}(\mathcal{K}))$$
$$|x| \mapsto \{C \in \text{sub}(\mathcal{K})) \mid x \in C^{\mathcal{I}}\}$$

This mapping is injective, which means that $|\Delta_f^{\mathcal{I}}| \leq |\mathcal{P}(\mathrm{sub}(\mathcal{K}))| = 2^{\mathrm{sub}}(\mathcal{K})$. Which proves Theorem 4.29.

Incompleteness

Lastly, we show that **Box** is not complete with respect to $\mathcal{EL}^{++}$ by considering the following system:

$$a : A \quad b : B \quad c : C$$
$$A \sqsubseteq AB \quad B \sqsubseteq BC \quad A \sqsubseteq AC$$
$$B \sqsubseteq AB \quad C \sqsubseteq BC \quad C \sqsubseteq AC$$
$$A \sqcap B \sqsubseteq \bot \quad B \sqcap C \sqsubseteq \bot \quad A \sqcap C \sqsubseteq \bot$$
$$AB \sqcap BC \sqsubseteq B \quad AC \sqcap BC \sqsubseteq C \quad AB \sqcap BC \sqsubseteq C$$

This knowledge base is satisfiable by the model $\{a, b, c\}$ with the interpretation:

$$A^{\mathcal{I}} = \{a\} \qquad B^{\mathcal{I}} = \{b\} \qquad C^{\mathcal{I}} = \{c\}$$
$$AB^{\mathcal{I}} = \{a, b\} \qquad BC^{\mathcal{I}} = \{b.c\} \qquad AC^{\mathcal{I}} = \{a, c\}$$

We would show that it has no box space model by assuming it does have one and showing that this leads to a contradiction.

Observe the following facts

Lemma 4.34 (Boxes are convex)**.** *Let B be a box in $S \subseteq \mathbb{R}^n$ and an element $z \in S$, if for any i there exist some $x, y \in B$ such that $x_i \leq z_i \leq y_i$, then $z \in B$.*

Proof. Denote the bounds of the box as $\vec{m}, \vec{M}$. For any i, since $x, y \in B$, we have that $m_i \leq x_i \leq z_i \leq y_i \leq M_i$ and $z_i \in [m_i, M_i]$. And so, $z \in B$. □

Assume for contradiction that this system can be represented using a box space $(S \subseteq \mathbb{R}^n, \mathcal{B})$. Denote the boxes by $B_\square := B(\vec{m^\square}, \vec{M^\square})$ where $\square \in \{A, B, C, AB, BC, AC\}$.

Claim 4.35. *If for a given j, we have $m_j^{AB} < m_j^A \leq M_j^A < M_j^{AB}$, then we must have $m_j^{AC} = m_j^A, M_j^{AC} = M_j^A$.*

Proof. First, observe that for any $i \in [n]$, we have $m_i^A \leq \frac{2m_i^A + (M_i^A - m_i^A)}{2} \leq M_i^A$. By convexity and the fact that B_A is a subset of both B_{AB} and B_{AC}, we find for all $i \in [n]$ that $\frac{2m_i^A + (M_i^A - m_i^A)}{2}$ is also in $[m_i^{AB}, M_i^{AB}]$ and $[m_i^{AC}, M_i^{AC}]$.

Let j be an index where $m_j^{AB} < m_j^A \leq M_j^A < M_j^{AB}$. Since $B_A \subseteq B_{AC}$, we have $m_j^{AC} \leq m_j^A \leq M_j^A \leq M_j^{AC}$. Assume that $m_j^{AC} < m_j^A$. We show that there exists a point x such that $x \in (B_{AB} \cap B_{AC}) \setminus B_A$. Thus, we reach a contradiction. Then one of two cases holds

1. $m_j^{AB} < m_j^{AC} < m_j^A$. By convexity, the point $x = (\frac{2m_i^A + (M_i^A - m_i^A)}{2} \chi_{i \neq j} + (\frac{2m_i^{AC} + (m_i^A - m_i^{AC})}{2} \chi_{i=j})_{i \in [n]}$ is both in AB and AC, but not in A.

2. $m_j^{AC} < m_j^{AB} < m_j^A$. By convexity, the point $x = (\frac{2m_i^A + (M_i^A - m_i^A)}{2}\chi_{i\neq j} + (\frac{2m_i^{AB} + (m_i^A - m_i^{AB})}{2}\chi_{i=j})_{i\in[n]}$ is both in AB and AC, but not in A.

Since we reached a contradiction in both cases, we conclude that $m_i^A = m_j^{AC}$. By a similar argument we have that $M_j^A = M_j^{AC}$. □

Similarly, if we have $m_j^{AB} < m_j^B \leq M_j^B < M_j^{AB}$, then we must have $m_j^{BC} = m_j^B, M_j^{BC} = M_j^B$.

Observe that for any $j \in [n]$, one of three cases holds:

i. $m_j^{AB} < m_j^A \leq M_j^A < M_j^{AB}$.

ii. $m_j^{AB} = m_j^A$.

iii. $M_j^A = M_j^{AB}$.

Likewise, B satisfies one of those three cases with respect to AB.

Claim 4.36. *Since $B_{AB} \cap B_{AC} = B_A$, for any i, we have $m_i^A = \max(m_i^{AB}, m_i^{AC})$ and $M_i^A = \min(M_i^{AB}, M_i^{AC})$.*

Since we must have $B_C \subseteq B_{AC} \cap B_{BC}$ and $B_B \subseteq B_{AB} \cap B_{BC}$, we can find a contradiction by showing $B_{AC} \subseteq S \setminus B_{BC}$ or $B_{AB} \subseteq S \setminus B_{BC}$. We shall show that there is an index k for which $M_k^{AC} < m_k^{BC}$, $M_k^{AB} < m_k^{BC}$ or $M_k^{AC} < m_k^{AB}$.

Since $B_A \cap B_B = \emptyset$, there exists some k for which either $M_k^A < m_k^B$ or $M_k^B < m_k^A$. Without loss of generality, assume that $M_k^A < m_k^B$ holds.

Then one of the following cases must be true

1. For both A and B, case (i) is not satisfied on k and AB. Notice that since $M_k^A < m_k^B$, we must have $m_k^A = m_k^{AB}$ and $M_k^B = M_k^{AB}$. This means that $M_k^{AC} = M_k^A < m_k^B = m_k^{AB}$ and we reached a contradiction.

2. Case (i) is satisfied for A on k but not for B on k and AB. This means that $M_k^B = M_k^{AB}$, and so $m_k^{BC} = m_k^B$ or $m_k^{AB} = m_k^B$. By Claim 4.35 on A, we also have $M_k^{AC} = M_k^A < m_k^B = m_k^{BC}$ or $M_k^{AC} = M_k^A < m_k^B = m_k^{AB}$. As in (1), we reached a contradiction.

3. Case (i) is satisfied for B on k but not for A on k and AB. This means that $m_k^A = m_k^{AB}$, and so $M_k^{AC} = M_k^A$ or $M_k^{AB} = M_k^A$. By Claim 4.35 on B, we also have $M_k^{AC} = M_k^A < m_k^B = m_k^{BC}$ or $M_k^{AB} = M_k^A < m_k^B = m_k^{BC}$. As in (1), we reached a contradiction.

4. Case (i) is satisfied for both A and B on k and AB. Then by Claim 4.35, we have $M_k^{AC} = M_k^A < m_k^B = M_k^{BC}$.

Since we reached a contradiction in all four cases, we showed that it cannot be the case that $M_k^A < m_k^B$. Similarly, we can show that it cannot be the case that $M_k^B < m_k^A$. Hence $B_A \cap B_B \neq \emptyset$. This is in contradiction to the assumption that the box space represents our knowledge base.

This proof implies that:

Theorem 4.37. **Box** *is incomplete with respect to* $\mathcal{EL}^{++}$.

In the previous section, we showed that every satisfiable $\mathcal{EL}^{++}$ knowledge base has a finite model. As to our knowledge, this result has not been shown previously in the literature. Our proof was an adaptation of a similar proof for $\mathcal{ALC}$ shown in [Baader et al., 2017]. We will later Theorem **??** to prove completeness of $\mathcal{EL}^{++}$ on a stronger class of embeddings.

Previous box embedding papers such as [Xiong et al., 2022b] and [Jackermeier et al., 2024a] have shown, in their word, that box spaces are *faithful* representations of $\mathcal{EL}^{++}$. E.g., given concepts B, C, D, if $B_B \cap B_C \subseteq B_D$ then $B \sqcap C \sqsubseteq D$.

However, none of the papers we have reviewed have acknowledged the incompleteness of box spaces against $\mathcal{EL}^{++}$. This fact implies that there are knowledge bases for which there does not exist a model that can faithfully represent them.

Chapter 5

Multibox Spaces

While **Box** is incomplete with respect to $\mathcal{EL}^{++}$, we can extend the category to a larger category that is complete with respect to $\mathcal{EL}^{++}$.

5.1 Multibox Category

Definition 5.1 (Multibox space)**.** *Given a box space $\mathcal{B}$, a multibox space $\mathcal{M} := \overline{\mathcal{B}}$ is the closure of $\mathcal{B}$ over finite unions and finite intersections. A subset of a multibox space $M \in \mathcal{M}$ is called a multibox. The subsets $\mathcal{M} \cap \mathcal{B}$ are called boxes.*

Many of the results in this section would rely on algebraic properties of sets. We would not prove most of these results but instead present them as they are. However, we would show the following

Lemma 5.2 (Sets are distributive)**.** *Given a finite set J, a collection of finite sets $\{I_j\}_{j\in J}$ and a collection of sets $\{A_{i,j}\}_{i\in I_j, j\in J}$, denote $F = \{f : J \to \bigsqcup_{j\in J} I_j \mid$ For any j we have $f(j) \in I_j\}$. Then we have that*

$$\bigcap_{j\in J}\bigcup_{i\in I_j} A_{i,j} = \bigcup_{f\in F}\bigcap_{j\in J} A_{f(j),j}$$

Proof. Given a set $\bigcap_{j\in J}\bigcup_{i\in I_j} A_{i,j}$, consider $F = \{f : J \to \bigsqcup_{j\in J} I_j \mid$ For any j we have $f(j) \in I_j\}$. Notice that $|F| = \prod_{j\in J}|I_j|$. As J and I_j for any $j \in J$ are finite, the right hand side of that expression is finite. Consider $\bigcup_{f\in F}\bigcap_{j\in J} A_{f(j),j}$.

Given $x \in \bigcap_{j\in J}\bigcup_{i\in I_j} A_{i,j}$, for any $j \in J$ there exists $i_x \in I_j$ such that $x \in A_{i_x,j}$. Fix a function $f \in F$ such that $f(j) = i_x$. This shows that there exists $f \in F$ such that for any $j \in J$ we have that $x \in A_{f(j),j}$. That is, $x \in \bigcup_{f\in F}\bigcap_{j\in J} A_{f(j),j}$.

□

Lemma 5.3. *Given a finite set J, a collection of finite sets $\{I_j\}_{j\in J}$ and a collection of* *sets $\{A_{i,j}\}_{i \in I_j, j \in J}$, there exists finite sets K such that*

$$\bigcup_{j \in J} \bigcup_{i \in I_j} A_{i,j} = \bigcup_{k \in K} A_k$$

Proof. The set $K = \bigsqcup_{j \in J} \{j\} \times I_j$ is finite and satisfies the requirement. □

Lemma 5.4. *Any multibox M is expressible as a finite union of boxes $M = \bigcup_{i \in I} B_i$.*

Proof. We would prove this result by induction on the construction of multiboxes. For the initial case, every box is expressible using itself.

Suppose that $M_1, \ldots, M_n$ were constructed in up to k steps. By induction hypothesis, those multiboxes are expressible using a finite union of boxes $M_j = \bigcup_{i \in I_j} B_{i,j}$ for any $j \in [n]$. Notice that $\bigcup_{j \in [n]} M_j = \bigcup_{j \in [n]} \bigcup_{i \in I_j} B_{i,j} = \bigcup_{k \in K} B_k$ for some finite set K. Thus the finite union of the multiboxes is expressible using a finite union of boxes.

Likewise, notice that $\bigcap_{j \in [n]} M_j = \bigcap_{j \in [n]} \bigcup_{i \in I_j} B_{i,j} = \bigcup_{k \in K} \bigcap_{l \in L} B_{l,k}$. As boxes are closed under finite intersection, we have that the right hand side of the expression is a finite union of boxes. Thus the finite intersection of the multiboxes is expressible using a finite union of boxes.

Thus we have shown that multiboxes that are constructed in up to $k+1$ steps can be expressible using finite union of boxes. By induction, this shows that any multibox is expressible using a finite union of boxes. □

Definition 5.5 (Multibox space morphism)**.** *Given two multibox spaces $(S, \mathcal{M})$ and $(S', \mathcal{M}')$, we define a multibox space morphism $f : (S, \mathcal{M}) \to (S', \mathcal{M}')$ as a function $f : S \to S'$ such that for all $M' \in \mathcal{M}'$ we have that $f^{-1}[M'] \in \mathcal{M}$.*

Like with boxes, multibox morphisms induce a meet semi-lattice morphism. We have a category **MultiBox** with multibox spaces as objects and multibox space morphisms as arrows.

Furthermore, we have a functor from the box category to the category of meet semi-lattices $U : \mathbf{MultiBox}^{op} \to \text{MeetSemiLat}$:

$$U_0 : (S, \mathcal{B}) \mapsto \mathcal{B}$$
$$U_1 : f(\cdot) \mapsto f^{-1}[\cdot]$$

Category of contexts

We begin by showing that **MultiBox** contains all finite products. We shall show this by proving that **MultiBox** has all binary products and a terminal object.

First observe that given two multibox spaces $(S, \bar{B}), (S', \overline{B'})$, since $\mathcal{B} \times \mathcal{B}'$ is a box space, we have that $\overline{\mathcal{B} \times \mathcal{B}'}$ is a multibox space.

Definition 5.6 (Product multibox space)**.** *Given two box spaces $(S, \mathcal{B})$ and $(S', \mathcal{B}')$, we define their product multibox space as $(S \times S', \overline{\mathcal{B} \times \mathcal{B}'})$. Given $\mathcal{M} = \overline{\mathcal{B}}, \mathcal{M}' = \overline{\mathcal{B}'}$, denote $\mathcal{M} \times \mathcal{M}' = \overline{\mathcal{B} \times \mathcal{B}'}$*

Lemma 5.7. *Given a set X and two sequences of subsets $\{A_i\}_{i\in I} \subseteq \mathcal{P}X$ and $\{B_j\}_{j\in J} \subseteq \mathcal{P}X$, we have that*

$$\bigcap_{i\in I} A_i \times \bigcap_{j\in J} B_j = \bigcap_{(i,j)\in I\times J} A_i \times B_j$$
$$\bigcup_{i\in I} A_i \times \bigcup_{j\in J} B_j = \bigcup_{(i,j)\in I\times J} A_i \times B_j$$

Lemma 5.8. *Given two multibox spaces $(S,\mathcal{M}),(S',\mathcal{M}')$ and multiboxes $M \in \mathcal{M}, M' \in \mathcal{M}'$ we have that $M \times M' \in \mathcal{M} \times \mathcal{M}'$.*

Proof. Expand $M = \bigcup_{i\in I} B_i, M' = \bigcap_{l\in L} B_l$

$$\begin{aligned} M \times M' &= \bigcup_{i\in I} B_i \times \bigcup_{l\in L} B_l \\ &= \bigcup_{(i,j)\in I\times L} B_i \times B_l \end{aligned}$$

As I, L are finite, $I \times L$ is finite. So the right hand side of the expression is a multibox. Thus, the left hand side of the expression is a multibox. □

Likewise, given $M \in \mathcal{M} \times \mathcal{M}'$, as there exist boxes $\{B_i\}_{i\in I} \in \mathcal{B} \times \mathcal{B}'$ such that $M = \bigcup_{i\in I} B_i$, and as every box B_i in the product space is a product of boxes, we have that all multiboxes of the product space are of the above form.

Before showing that the product box space is the product in **MultiBox**, observe the following result

Lemma 5.9 (The preimage preserves unions and intersections)**.** *Given any function $f : X \to Y$ and subsets $\{S_i\}_{i\in I}$ of Y, we have that*

$$f^{-1}\left[\bigcup_{i\in I} S_i\right] = \bigcup_{i\in I} f^{-1}[S_i]$$
$$f^{-1}\left[\bigcap_{i\in I} S_i\right] = \bigcap_{i\in I} f^{-1}[S_i]$$

Lemma 5.10. *The product multibox space is the product in* **MultiBox**.

Proof. Given a projection $(S\times S', \mathcal{M}\times\mathcal{M}')$ and multibox $M = \bigcup_{i\in I} B_i \in \mathcal{M}$, notice that $\pi^{-1}[M] = \bigcup_{i\in I} B_i \times S'$ which is a multibox.

Given a multibox space $(A,\mathcal{A})$ and two multibox morphisms $f : (A,\mathcal{A}) \to (S,\mathcal{M})$ and $g : (A,\mathcal{A}) \to (S',\mathcal{M}')$, the unique map of sets $(f,g) : x \to (f(x), g(x))$

that commutes with the projections is also a multibox morphism. This is because for any $M := \bigcup_{i\in I} B_i^1 \times B_i^2 \in \overline{\mathcal{B}\times\mathcal{B}'}$, we have that

$$\begin{aligned}(f,g)^{-1}[M] &= (f,g)^{-1}\left[\bigcup_{i\in I} B_i^1 \times B_i^2\right] \\ &= \bigcup_{i\in I}(f,g)^{-1}[B_i^1 \times B_i^2] \\ &= \bigcup_{i\in I} f^{-1}[B_i^1]\times g^{-1}[B_i^2]\end{aligned}$$

As f and g are multibox morphisms, the preimage of both of them is a multibox in $\mathcal{A}$. Thus the final expression is a multibox in $\overline{\mathcal{S}}$. Thus, (f,g) is a multibox morphism. □

Notice that every pair of box spaces has a binary product.

Lemma 5.11 (The terminal object)**.** *The box space* $(\{0\},\{\varnothing,\{0\}\})$ *is the terminal object in* **MultiBox**.

By Lemma Theorem 3.20, these results imply that

Lemma 5.12. **MultiBox** *has all finite products.*

Left adjoint

Definition 5.13 (Existential quantifier in **MultiBox**)**.** *Given a projection* $\pi : (S\times S', \mathcal{M}\times\mathcal{M}') \to (S,\mathcal{M})$, *we define the existential quantifier as* $\exists_\pi = \pi$

It is clear that the existential quantifier is a functor.

Lemma 5.14 (Existential quantifier is a left adjoint)**.** *For any projection*

$$\pi : (S\times S', \mathcal{M}\times\mathcal{M}') \to (S,\mathcal{M})$$

the existential operator $\exists_\pi$ *is the left adjoint of* π^{-1} *in* $\text{MeetSemiLat}_{\leq}$.

As the proof of left adjunction relies on the properties of the projection on the underlying sets, the proof of the lemma is similar to the proof of left adjunction in **Box**.

And likewise

Corollary 5.15. *For any morphism* $\pi : (S,\mathcal{M}) \to (S',\mathcal{M}')$ *and objects* $M\in\mathcal{M}$ *and* $M'\in\mathcal{M}'$, *we have that* $\exists_\pi \circ \pi^{-1} M \subseteq M$ *and* $M' \subseteq \pi^{-1}\circ\exists_\pi M'$

Remark 5.16 (Existential quantifier and the product box space)**.** *Given a product multibox space and a projection* $\pi : (S\times S', \mathcal{M}\times\mathcal{M}') \to (S,\mathcal{M})$, *we have in particular that*

$$\exists_\pi(M\times M') = \{\vec{x} \mid \exists \vec{x'} \in \mathcal{M}' : (\vec{x},\vec{x'}) \in M\times M'\}$$

We need to show that the existential operator satisfies the Beck-Chevalley condition and the Frobenius reciprocity.

However, the proofs of both conditions are exactly like the proofs for the **Box** case.

Lemma 5.17 (Beck-Chevalley condition for **MultiBox**). *Given any three multibox spaces $(A, \mathcal{A}), (C, \mathcal{C})$ and $(D, \mathcal{D})$, a map $f : (C, \mathcal{C}) \to (D, \mathcal{D})$ and pullback diagram*

$$\begin{array}{ccc} (A \times C, \mathcal{A} \times \mathcal{C}) & \xrightarrow{\pi} & (C, \mathcal{C}) \\ \downarrow{\scriptstyle (Id_A, f)} & & \downarrow{\scriptstyle f} \\ (A \times D, \mathcal{A} \times \mathcal{D}) & \xrightarrow{\pi'} & (D, \mathcal{D}) \end{array}$$

in **MultiBox**, *the Beck-Chevalley condition holds in* MeetSemiLat.

$$\exists_\pi \circ (Id_A, f)^{-1} = f^{-1} \circ \exists_{\pi'}$$

Lemma 5.18 (Frobenius reciprocity for **MultiBox**). *Given a projection multibox space morphism $\pi : (A \times B, \mathcal{A} \times \mathcal{B}) \to (A, \mathcal{A})$, the existential operator $\exists_\pi$ satisfies the Forbenius reciprocity. That is, for any two multiboxes $M \times C \in \mathcal{A} \times \mathcal{B}$ and $M' \in \mathcal{A}$,*

$$\exists_\pi(\pi^{-1}(M') \cap (M \times C)) = M' \cap \exists_\pi(M \times C)$$

Combining the two lemmas, we can conclude that

Theorem 5.19 (Multibox spaces form a hyperdoctrine). *The functor U :* **MultiBox** $\to$ **MeetSemiLat** *is a $\mathcal{EL}^{++}$ hyperdoctrine.*

5.2 Multibox Interpetations

The categorical interpretation $[\![\cdot]\!]$ of $\mathcal{EL}^{++}$ in **MultiBox** is exactly the same as in **Box** with concepts and individuals being mapped to multiboxes.

5.2.1 Model Theoretic Results

Theorem 5.20. *Given a finite model $(\Delta^\mathcal{I}, \cdot^\mathcal{I})$ of a knowledge base $\mathcal{K}$, there exists a multibox space that interprets $\mathcal{K}$.*

Proof. Consider an enumeration $\Delta^\mathcal{I} = \{a_1, \dots, a_n\}$. The underlying set would be $[0, n]$ and for every individual a_i fix a box $A_i = [n - \frac{1}{3}, n + \frac{1}{3}]$. For every strict subset $C \subset \Delta^\mathcal{I}$, define $M_C = \bigcup_{i \in \{i | a_i \in C\}} A_i$ and define also $M_{\Delta^\mathcal{I}} = [0, n]$. Likewise, we construct for every subset $R \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$ a corresponding product multibox $M_R = \bigcup_{i,j \in [n], (a_i, a_j) \in R} A_i \times A_j$.

Notice that for every two subsets $C, D \subseteq \Delta^{\mathcal{I}}$, we have that $[\![C]\!] \cap [\![D]\!] = M_C \cap M_D = M_{C\cap D} = [\![C \cap D]\!]$. Also,

$$\begin{aligned} M_{\exists R.C} &= M_{\{a\in\Delta^{\mathcal{I}} \mid \exists b.(a,b)\in R\wedge b\in C\}} \\ &= \bigcup_{i\in\{i \mid \exists b.(a_i,b)\in R\wedge b\in C\}} A_i \end{aligned}$$

It is easy to see that $\bigcup_{(i,j)\in\{(i,j)\mid(a_i,b_j)\in R\cap(S\times C)\}} A_i \times A_j \subseteq M_R \cap (S \times M_C)$. Let $x \in R \cap (S \times C)$. Since $x \in R$, there exists A_i, A_j, A_k such that $a_k \in C, (a_i, a_j) \in R$ and also $x \in A_i \times A_j, x \in S \times A_k$. If $j \neq k$, then $A_i \cap A_j = \emptyset$ which would result in a contradiction. Which implies that $a_j = a_k \in C$. Thus $x \in M_R \cap (S \times M_C)$ which implies that the two sets are equal to one another. This means that:

$$\begin{aligned} M_{\exists R.C} &= \pi_0[\bigcup_{(i,j)\in\{(i,j)\mid(a_i,b_j)\in R\cap(S\times C)\}} A_i \times A_j] \\ &= \pi_0[M_R \cap (S \times M_C)] = \pi_0[M_R \cap \pi_1^{-1}[M_C]] = [\![\exists R.C]\!] \end{aligned}$$

Lastly, we have that

$$\begin{aligned} a_i \in C &\Leftrightarrow A_i \in M_C \\ C \subseteq D &\Leftrightarrow M_C \subseteq M_D \end{aligned}$$

Due to that, the following is an interpretation of $\mathcal{K}$ in $(S, \mathcal{M})$

$$\begin{aligned} [\![a]\!] &:= A_i \quad \text{where } a^{\mathcal{I}} = a_i \\ [\![C]\!] &:= M_{C^{\mathcal{I}}} \\ [\![R]\!] &:= M_{R^{\mathcal{I}}} \end{aligned}$$

If a formula $\varphi \sqsubseteq \psi$ (including assertions $\{a\} \sqsubseteq \psi$) is contained in $\mathcal{K}$, then $[\![\varphi]\!] = M_\varphi \subseteq M_\psi = [\![\psi]\!]$. Furtheremore, any point $x_{a^{\mathcal{I}}}$ is itself a box.

Lastly, it is easy to see that any box of an individual is atomic and that for any individual $[\![\exists x : \sigma.\{a\}]\!] = \pi[\{x_{a^{\mathcal{I}}}\}] = \top$. □

Combining Theorems **??** and 5.20, we get

Theorem 5.21. *$\mathcal{EL}^{++}$ is complete with respect to* **MultiBox**. *That is, if a $\mathcal{EL}^{++}$ knowledge base is satisfiable, then it has a multibox interpretation.*

5.3 Universal Quantification

The fact that multiboxes are closed under finite unions suggests that there is also a functor from a multibox space to a lattice. Furthermore, in this section we would show that the projections have a right adjoint that is a universal quantifier. This would suggest that, assuming multiboxes also form a Heying algebras, there might exist an interpretation of $\mathcal{ALC}$ using multibox spaces.

However, note that the negation of a multibox lattice would not in general be the complement $(\cdot)^C$. This would imply that for any object M, we would probably not have $M \vee \neg M$ It is easy to show that $\mathcal{ALC}$ knowledge bases all contain the law of excluded middle. Which would mean that $\mathcal{ALC}$ is probably not complete with respect to this system.

In the following chapter we would introduce quasi-multiboxes, a system that is complete against $\mathcal{ALC}$. However, since it is easier to work with multiboxes, we would show that multiboxes have a universal quantifier and deduce in the next chapter that this would imply that quasi-multiboxes also have a universal quantifier.

Definition 5.22 (Box operator)**.** *Given a multibox product space $(A, \mathcal{M}) \times (B, \mathcal{M}')$ and projection $\pi : A \times B \to A$, we define the box operator $\square : \mathcal{P}(A \times B) \to \mathcal{P}A$ as:*

$$\square U := A \setminus \pi[(A \times B) \setminus U] = (\pi[U^C])^C$$

We would show that $\square$ is a right adjoint to π and then show that the right adjoint is a universal quantification in the hyperdoctrine.

5.3.1 Box operator is a right adjoint

Box operator is a functor

Lemma 5.23. *Given a multibox projection $\pi_S : (S \times S', \mathcal{M} \times \mathcal{M}') \to (S, \mathcal{M})$, we have for any multiboxes $M \in \mathcal{M} \times \mathcal{M}'$ that $\square M = \{x \in S \mid$ for all $y \in S'$ we have that $(x, y) \in M\} =: K_M$.*

Proof. Given $x \in K_M$, there does not exist y such that $(x, y) \notin M$. Hence for all y we have that $(x, y) \notin M^C$. Which implies that $x \notin \pi[M^C]$. As x was chosen arbitrarily, we see that $K_M \subseteq \square M$.

Suppose that $x \notin K_M$. Then there exists $y \in S'$ such that $(x, y) \notin M$. Then $(x, y) \in M^C$. Meaning that $x \in \pi[M^C]$ which implies that $x \notin \square M$. This implies that $K_M^C \subseteq (\square M)^C$. Meaning that $\square M \subseteq K_M$. □

Lemma 5.24. *For any multibox M, we have that K_M is a multibox.*

Proof. Expand $M = \bigcup_{i \in I} B_i^1 \times B_i^2$. Define $\mathcal{A}_M = \{A \subseteq I \mid \bigcup_{i \in A} B_i^2 = S'\}$ and consider

$$\bigcup_{A \in \mathcal{A}_M} \bigcap_{i \in A} B_i^1$$

Notice that each A is finite and so $\bigcap_{i \in A} B_i^1$ is finite. As boxes are closed under finite intersection $\bigcap_{i \in A} B_i^1$ is a box. Furthermore, $|\mathcal{A}| \leq |\mathcal{P}I| = 2^{|I|}$. As the right hand side of the equation is finite, $\mathcal{A}$ is finite and so the expression is a finite union of boxes and thus a multibox.

$\subseteq$ Let $x \in K_M$. Define the set $A := \{i \in I \mid$ For some $y \in S'$ we have that $(x, y) \in B_i^1 \times B_i^2\}$. As $x \in K_M$, we have that $\bigcup_{i \in A} B_i^2 = S'$. Which means that $A \in \mathcal{A}_M$. Furthermore, by definition we have for any $i \in A$ that $x \in B_i^1$. This implies that $x \in \bigcup_{A \in \mathcal{A}_M} \bigcap_{i \in A} B_i^1$.

$\supseteq$ Suppose that $x \in \bigcup_{A \in \mathcal{A}_M} \bigcap_{i \in A} B_i^1$. Then there exists $A \in \mathcal{A}_M$ such that for any $i \in A$ we have that $x \in B_i^1$. Furthermore, as $\bigcup_{i \in A} B_i^2 = S'$, we have that for any $y \in S'$, there exists $i \in A$ such that $(x,y) \in B_i^1 \times B_i^2 \subseteq M$. Thus $x \in K_M$. This shows that $K_M = \bigcup_{A \in \mathcal{A}_M} \bigcap_{i \in A} B_i^1$. □

Lemma 5.25 (The box operator is monotone)**.** *Given two boxes $M, M' \in \mathcal{M} \times \mathcal{M}'$, if $M \subseteq M'$ then $\Box M \subseteq \Box M'$.*

Proof.

$$\begin{aligned} M \subseteq M' &\Rightarrow (M')^C \subseteq M^C \\ &\Rightarrow \pi[(M')^C] \subseteq \pi[M^C] \\ &\Rightarrow (\pi[M^C])^C \subseteq (\pi[(M')^C])^C \end{aligned}$$

□

As the box operator is both monotone and preserves boxes, it is a functor between meet lattices.

Box operator is right adjoint to the preimage

Theorem 5.26. *Box operator is a right adjoint to the preimage.*

Proof. Let $M := \bigcup B \in \mathcal{M}$ and $K \in \mathcal{M} \times \mathcal{M}'$. We shall show that

$$M \times S' = \pi^{-1}(M) \subseteq K \text{ if and only if } M \subseteq \Box K$$

$\Rightarrow$ Suppose that $M \times S' \subseteq K$. Let $\vec{x} \in M$ and suppose that $\vec{x} \in \pi[K^C]$. That is, there exists $\vec{y} \in S'$ such that $(\vec{x}, \vec{y}) \in K^C$. But since $(\vec{x}, \vec{y}) \in M \times S'$, we also have that $(\vec{x}, \vec{y}) \in K$. This is a contradiction, which implies that $\vec{x} \notin \pi[K^C]$. As $\vec{x}$ was chosen arbitrarily, we get that $M \subseteq (\pi[K^C])^C = \Box K$.

$\Leftarrow$ Denote $K := \bigcup_{i \in I} B_i \times B_i'$ Assume that $M \subseteq \Box K = \Box(\bigcup_{i \in I} B_i \times B_i') = \{x \in S \mid \text{ for all } y \in S' \text{ we have that } (x,y) \in K\}$. Given $(\vec{x}, \vec{y}) \in M \times S'$, we have that $\vec{x} \in \{x \in S \mid \text{ for all } y \in S' \text{ we have that } (x,y) \in K\}$. And so $(x,y) \in K$. □

5.3.2 Box operator is a universal quantifier

Denote the box operator $\Box$ on projection π by $\forall_\pi$.

Lemma 5.27 (Beck-Chevalley condition for box operator in **MultiBox**)**.** *Given any three multibox spaces $(A, \mathcal{A}), (C, \mathcal{C})$ and $(D, \mathcal{D})$, a map $f : (C, \mathcal{C}) \to (D, \mathcal{D})$ and pullback diagram*

$$\begin{array}{ccc} (A \times C, \mathcal{A} \times \mathcal{C}) & \xrightarrow{\ \pi\ } & (C, \mathcal{C}) \\ \Big\downarrow{\scriptstyle (Id_A, f)} & & \Big\downarrow{\scriptstyle f} \\ (A \times D, \mathcal{A} \times \mathcal{D}) & \xrightarrow{\ \pi'\ } & (D, \mathcal{D}) \end{array}$$

in **MultiBox**, *the Beck-Chevalley condition holds in* MeetSemiLat.

$$\forall_{\pi}(Id_A, f)^{-1} = f^{-1}\forall_{\pi'}$$

As the proof relies solely on set theoretic properties, we would not show it here.

As $\forall_{\pi}$ is a functor that satisfies the Beck-Chevalley condition, we can conclude that it is the universal quantifier in our underlying category.

Chapter 6

Quasi-Multibox Sets

We would show that $\mathcal{ALC}$ is interpretable in a modified version of multibox spaces. For this task, we would construct a functor from multibox spaces to the category of Boolean algebras and show that $\mathcal{ALC}$ is complete with respect to that system.

6.1 The $\mathcal{ALC}$ Fragment of First-Order Logic

Definition 6.1 ($\mathcal{ALC}$ Language)**.** *Given a signature $(N_\mathcal{C},\mathcal{R},\mathcal{O})$, as defined in definition (1.1), the corresponding $\mathcal{ALC}$ language is $(\{\sigma\},\mathcal{R}\cup N_\mathcal{C},\mathcal{O})$.*

Where the arity of any relation symbol $R \in \mathcal{R}$ is (σ,σ), the arity of any concept symbol $C \in N_\mathcal{C}$ is σ and the arity of any individual $a \in \mathcal{O}$ is $() \to \sigma$

Definition 6.2 ($\mathcal{ALC}$ Formula)**.** *An $\mathcal{ALC}$ formula φ in context Γ is defined recursively by the following grammar:*

$$\top \mid \bot \mid x =_\sigma y \mid R(t_1,t_2) \mid C(t) \mid \psi \wedge \chi \mid \psi \vee \chi \mid \exists x:\sigma.R(x,t) \wedge \varphi(t) \mid \forall x:\sigma.R(x,t) \wedge \varphi(t) \mid \neg\varphi$$

where φ,ψ are previously constructed formulas, x,y are terms in context Γ, a is a constant symbol, R is a relation symbol and C is a concept symbol.

The rules for term and formula substitution are as found in definitions 3.50, 3.53.

Lastly, notice that this system has only two types of terms: fixed terms $a:\sigma$ and variables $x:\sigma$.

6.2 Quasi-Multiboxes are a Boolean Algebra

Notice that any box B in n dimensions can be represented as a Cartesian product of intervals $B = \bigtimes_{j\in[n]} I_j$ where I_j are intervals of the form $[m_j,M_j]$.

Definition 6.3 (Quasi-Box and Almost Intervals). *We define almost intervals $\tilde{I}_j$ interval as sets of the form $(m_j, M_j], [m_j, M_j], [m_j, M_j)$.*

We define a quasi-box as a set consisting of a cartesian product of almost intervals $\tilde{B} = \bigtimes_{j\in[n]} \tilde{I}_j$.

Lemma 6.4 (Quasi-boxes are closed under intersection). *Given a collection $\{\tilde{B}_k\}_{k\in K}$ where K is arbitrary, we have that $\bigcap_{k\in K} \tilde{B}_k$ is a quasi-box.*

Proof. It suffices to notice that any intersection of a set of almost intervals is almost an interval and that $\bigcap_{k\in K} \tilde{B}_k = \bigcap_{k\in K} \tilde{I}_{j,k} = \bigtimes_{j\in[n]} \bigcap_{k\in K} \tilde{I}_{j,k}$. □

Definition 6.5 (Quasi-box space). *Given a set I and a set S, we call $I \cap S$ the set I restricted to S.*

We define a quasi-box space as a subset $S \subseteq \mathbb{R}^n$ and a collection of quasi-boxes $\tilde{\mathcal{B}}$ restricted to S closed under arbitrary intersections and containing $\varnothing$ and S.

Definition 6.6 (Quasi-multibox). *We define a quasi-multibox as a set $\tilde{M}$ that can be constructed using a finite union of quasi-boxes.*

Lemma 6.7 (The complement of a quasi-box is quasi-multibox). *Given $\tilde{B}$, there exists a multibox $\tilde{M}$ such that $\tilde{B}^C = \tilde{M}$.*

Proof. Notice that for any almost interval $\tilde{I}$, we have that the complement $\tilde{I}^C$ is a union of at most two almost intervals $\tilde{I}^C = \tilde{I}_1 \cup \tilde{I}_2$. Meaning that

$$\begin{aligned}
\tilde{B}^C &= (\bigtimes_{j\in[n]} \tilde{I}_j)^C \\
&= \bigcup_{A\subsetneq[n]} \{\vec{x} \,|\, \text{for } j \in A \text{ we have } x_j \in \tilde{I}_j \text{ and for } j \notin A \text{ we have } x_j \in \tilde{I}_j^C\} \\
&= \bigcup_{A\subsetneq[n]} \{\vec{x} \,|\, \text{for } j \in A \text{ we have } x_j \in \tilde{I}_j \text{ and for } j \notin A \text{ we have } x_j \in \tilde{I}_{1,j} \cup \tilde{I}_{2,j}\} \\
&= \bigcup_{A\subsetneq[n]} \{\vec{x} \,|\, \text{for } j \in A \text{ we have } x_j \in \tilde{I}_j \text{ and for } j \notin A \text{ we have } x_j \in \tilde{I}_{1,j}\} \\
&\qquad \cup \{\vec{x} \,|\, \text{for } j \in A \text{ we have } x_j \in \tilde{I}_j \text{ and for } j \notin A \text{ we have } x_j \in \tilde{I}_{2,j}\}
\end{aligned}$$

Notice that the right hand side is a finite union of quasi-multiboxes and so the left hand side is a quasi-multibox. □

Lemma 6.8 (The complement of a quasi-multibox is a quasi-multibox). *Given $\tilde{M}$, there exists a multibox $\tilde{M}'$ such that $\tilde{M}^C = \tilde{M}'$.*

Proof. Notice that

$$\tilde{M}^C = (\bigcup_{i\in I} \tilde{B}_i)^C = \bigcap_{i\in I} \tilde{B}_i^C$$

As the right hand side is a finite intersection of quasi-boxes, the left hand side is a a quasi-multibox. □

Definition 6.9 (Quasi-multibox space)**.** *We define quasi-multibox space as the closure of a quasibox space under finite unions, intersections and complements* $\tilde{\mathcal{M}} = \tilde{\mathcal{B}}$.

Theorem 6.10. *Every quasi-multibox space* $(S, \tilde{\mathcal{M}})$ *is a Boolean algebra as a poset under subsets and under the operations*

$\top$	S
$\bot$	$\varnothing$
$M \wedge N$	$M \cap N$
$M \vee N$	$M \cup N$
$M \to N$	$M^C \cup N$

Proof. To show that the space is a Heyting algebra, it suffices to observe that for any $M, N, L \in \tilde{\mathcal{M}}$, we have that

- $\varnothing \subseteq M \subseteq S$.
- $L \subseteq M \cap N$ if and only if $L \subseteq M$ and $L \subseteq N$.
- $M \cup N \subseteq L$ if and only if $M \subseteq L$ and $N \subseteq L$.
- $L \subseteq M^C \cup N$ if and only if $L \cap M \subseteq N$.

Lastly, notice that $\neg M = M \to \varnothing = M^C$ and that for any M we have that $M \cup M^C = S$. □

6.2.1 The Quasi-Multibox Hyperdoctrine

Definition 6.11. *Given two quasi-multibox spaces* $(S, \tilde{\mathcal{M}})$ *and* $(S', \tilde{\mathcal{M}}')$, *we define a quasi-multibox morphism as a function* $f : S \to S'$ *such that for any* $A \in \tilde{\mathcal{M}}'$ *we have that* $f^{-1}[A] \in \tilde{\mathcal{M}}$.

Notice that due to the fact that preimages in general preserve unions, intersections and complements, the quasi-multibox morphisms is also a Boolean algebra morphism.

It is easy to show that the morphisms are composable and associative and that the identity function is a quasi-multibox morphism.

Definition 6.12 (Quasi multibox category)**.** *We define the quasi-multibox category* **QuaMultiBox** *as the category with quasi-multibox spaces as objects and quasi-multibox morphisms as morphisms.*

Lemma 6.13. *There exists a forgetful functor from the category of quasi-multiboxes to the category of Boolean algebras* $U : \mathbf{QuaMultiBox}^{\mathrm{op}} \to \mathrm{BoolAlg}$ *such that* $(S, \tilde{M}) \mapsto \tilde{M}$ *and* $f \mapsto f^{-1}$.

Definition 6.14 (Product of quasi-multibox spaces)**.** *Given two quasi-multibox spaces* $(S, \tilde{\mathcal{M}})$ *and* $(S', \tilde{\mathcal{M}}')$, *we define their product quasi-multibox space* $\tilde{\mathcal{M}} \times \tilde{\mathcal{M}}'$ *as* $(S \times S', \overline{\tilde{\mathcal{B}} \times \tilde{\mathcal{B}}'})$

By the same reasoning as in the multibox case, the quasi-multiboxes has finite products and terminal object and so form a category of contexts.

Furthermore, given a projection $\pi : S \times S' \to S$, the left and right adjoints to π^{-1} are

$$\exists_\pi = \pi$$
$$\forall_\pi = (\pi[\cdot^C])^C$$

and they satisfy the Beck-Chevalley condition and the Frobenius reciprocity. Meaning that

Theorem 6.15 (Quasi-multibox spaces form a hyperdoctrine)**.** *The functor* **QuaMultiBox** $\to$ BoolAlg *is a first-order hyperdoctrine.*

6.3 Quasi-Multibox Interpretations

Definition 6.16 (Categorical Semi-Interpretation of $\mathcal{ALC}$ in **QuaMultiBox**)**.** *Given an $\mathcal{ALC}$ language $(\{\sigma\}, \mathcal{R} \cup N_\mathcal{C} \cup \mathcal{O}, \varnothing)$, a quasi-multibox space semi-interpretation of the language is a quasi-multibox space $(S, \overline{\mathcal{M}})$, along with a mapping*

$$[\![\cdot]\!] : \mathcal{C} \to \overline{\mathcal{M}}$$
$$[\![\cdot]\!] : \mathcal{R} \to \overline{\mathcal{M}} \times \overline{\mathcal{M}}$$
$$[\![\cdot]\!] : \mathcal{O} \to \overline{\mathcal{M}} \setminus \{\bot\}$$

Given a formula φ in context σ, the interpretation of φ is defined recursively on σ as:

1. *If φ is $\top$ or $\bot$ then $[\![\varphi]\!]$ is S or $\varnothing$ respectively.*
2. *If $\varphi := \psi \wedge \chi$, then $[\![\varphi]\!] = [\![\psi \wedge \chi]\!] := [\![\varphi]\!] \cap [\![\psi]\!]$.*
3. *If $\varphi := \psi \vee \chi$, then $[\![\varphi]\!] = [\![\psi \wedge \chi]\!] := [\![\varphi]\!] \cup [\![\psi]\!]$.*
4. *If $\varphi := \neg\psi$, then $[\![\varphi]\!] = [\![\neg\psi]\!] := ([\![\varphi]\!])^C$.*
5. *If $\varphi := \exists x : \sigma.\psi$ where ψ is in context $x : \sigma, y : \sigma$ with projections π_1 and π_2, then $[\![\varphi]\!] = [\![\exists x : \sigma.\psi]\!] := \pi_1([\![\psi]\!])$.*

Definition 6.17 (Categorical Interpretation of $\mathcal{ALC}$)**.** *Given a $\mathcal{ALC}$ semi-interpetation $((S, \overline{\mathcal{M}}), [\![\cdot]\!])$ is a $\mathcal{ALC}$ interpretation if it satisfies the following additional condition*

- *If $[\![\psi]\!] \subseteq [\![\{a\}]\!]$ for some $\psi \neq \bot$, then $[\![\{a\}]\!] \subseteq [\![\psi]\!]$.*

Recall that $\exists R.C$ is read as $\exists y : \sigma.R \wedge C$. That is

$$[\![\exists y : \sigma.R(x,y) \wedge C(y)]\!] = Cl \circ \pi_1([\![R(x,y) \wedge C(y)]\!]) = Cl \circ \pi_1([\![R]\!] \cap \pi_2^{-1}[\![C]\!])$$

6.3.1 Model Theoretic Results

Theorem 6.18 ([Baader et al., 2017]). *Every satisfiable knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ of $\mathcal{ALC}$ has a finite model.*

Theorem 6.19. *Given a finite model $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of a knowledge base $\mathcal{K}$, there exists a quasi-multibox space that interprets $\mathcal{K}$.*

Proof. Consider an enumeration $\Delta^{\mathcal{I}} = \{a_1, \ldots, a_n\}$. The underlying set would be $[0, n]$, for every $i \in [1, n-1]$ fix a box $A_i = [n - \frac{1}{3}, n + \frac{1}{3}]$ and lastly fix $A_n = \bigcap A_i^C$. For every subset $C \subseteq \Delta^{\mathcal{I}}$, define $M_C = \bigcup_{i \in \{i | a_i \in C\}} A_i$. Likewise, we construct for every subset $R \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ a corresponding product multibox $M_R = \bigcup_{i,j \in [n], (a_i, a_j) \in R} A_i \times A_j$.

Notice that for every two subsets $C, D \subseteq \Delta^{\mathcal{I}}$, we have that $M_C \cap M_D = M_{C \cap D}$ and $M_C \cup M_D = M_{C \cup D}$.

Notice that for any i we have that $A_i^C = \bigcup_{j \in [n] \setminus \{i\}} A_j$. Hence

$$(M_C)^C = \bigcap_{\{i | a_i \in C} A_i^C = \bigcap_{\{i | a_i \in C\}} A_i^C = \bigcap_{\{i | a_i \in C\}} \bigcup_{j \in [n] \setminus \{i\}} A_j = \bigcup_{\{i | a_i \notin C\}} A_i = M_{C^C}$$

Lastly, $M_{\exists R.C} = \exists R.(M_C)$ and $M_{\forall R.C} = M_{\{a \in \Delta^{\mathcal{I}} | (a,b) \notin R \vee b \in C\}} = \mathrm{dom}(M_{R^C \cup S \times C}) = \mathrm{dom}(M_R^C \cup M_{S \times C}) = \forall R.(M_C)$

defines a multibox. Lastly, we have that

$$a_i \in C \Leftrightarrow A_i \in M_C$$
$$C \subseteq D \Leftrightarrow M_C \subseteq M_D$$

Due to that, the following is an interpretation of $\mathcal{K}$ in $(S, \mathcal{M})$

$$[\![a]\!] := A_i \quad \text{where } a^{\mathcal{I}} = a_i$$
$$[\![C]\!] := M_{C^{\mathcal{I}}}$$
$$[\![R]\!] := M_{R^{\mathcal{I}}}$$

If a formula $\varphi \sqsubseteq \psi$ (including assertions $\{a\} \sqsubseteq \psi$) is contained in $\mathcal{K}$, then $\varphi^{\mathcal{I}} \subseteq \psi^{\mathcal{I}}$. Which implies that $M_{\varphi^{\mathcal{I}}} \subseteq M_{\psi^{\mathcal{I}}}$. Furtheremore, any point $x_{a^{\mathcal{I}}}$ is itself a box.

Lastly, it is easy to see that any box of an individual is atomic and that for any individual $[\![\exists x : \sigma.\{a\}]\!] = \pi[\{x_{a^{\mathcal{I}}}\}] = \top$. □

Combining Theorems 6.18 and 6.19, we get

Theorem 6.20. *$\mathcal{ALC}$ is complete with respect to* **QuaMultiBox**.

Part III

Experiments

Chapter 7

Experimental Setup

We have shown that multibox spaces are sound and complete with respect to $\mathcal{EL}^{++}$. This means that it is possible to find an accurate representation of every knowledge base written in $\mathcal{EL}^{++}$ using a multibox space. However, the question arises whether they are efficient for knowledge base completion tasks.

That is, can we produce a representation that not only accurately captures the structure of the knowledge base but also allows for efficient inference and prediction of missing relationships? To address this question, we design experiments that test the practical performance of multibox spaces in knowledge base completion tasks. We would use real-world knowledge bases written in $\mathcal{EL}^{++}$ and compare box and multibox representations on the missing relationship inference task.

To construct the representations, we would use a stochastic gradient descent algorithm. We would initialize a random representation of the basic concepts and roles in the knowledge base as (multi)boxes and iteratively adjust the representations by minimizing a loss function that enforce the subsumption relations $C \sqsubseteq D$.

The representations would then be evaluated against the evaluation data of each dataset to test its usefulness for knowledge completion by using the Hit@K metric.

7.1 Multibox Model

To test our new embedding, we implemented the MultiboxEL model [Aharoni, 2024]. Given a signature $\Sigma = (N_{\mathcal{C}}, \mathcal{R}, \mathcal{O})$ and hyperparameters

- d for the dimensionality of the space.
- n for the amount of boxes per multibox.
- bo for bounds of the space (that is, our space would be a box $S := [-bo, bo]^d$).

- $\#QMC$ the amount of points to initialize for our quasi-Monte Carlo algorithm, see comment below.

we define the model M such that the weights of the model are three tensors A, M, R of the form $|O| \times 1 \times 2d$, $|N_{\mathcal{C}}| \times n \times 2d$,$|R| \times n \times 4d$ correspondingly. For each box B_M of each multibox M, the first half represents the centre of the box $\vec{c}_{B_M}$ and the second half represents the offset $\vec{l}_{B_M}$. That is, a box is represented as $B = (c\vec{_{B_M}} - l\vec{_{B_M}}, c\vec{_{B_M}} + l\vec{_{B_M}})$. We use center-offset instead of minimum-maximum as it makes it easier to initialize the weights such that all boxes are non-zero.

Our model receives as input subsets of rules of a knowledge base $\mathcal{K}$ written in Σ and outputs a score of the distance of the current embedding from satisfying these rules.

7.1.1 Loss Function and Monte-Carlo Method

In the box embedding literature, the loss function equation is standardized and is measured by the level of subsumption of box B_1 by box B_2 using the center and overlap of each box ([Kulmanov et al., 2019b]).

$$\mathcal{L}_{A \sqsubseteq B} = 1 - \frac{\text{Vol}([\![A]\!] \cap [\![B]\!])}{\text{Vol}([\![A]\!])}$$

Measuring the volume of a union of hyper dimensional boxes is known as Klee's measure problem [Klee, 1977] and has an upper bound complexity of $O(n^{d/2})$ where n is the amount of rectangles and d is the amount of dimensions [Overmars and Yap, 1991].

Due to the high complexity of this problem and difficulty of implementing a solution, we instead approximated the volume by using a Monte Carlo algorithm. At the beginning of the training, we uniformly distribute across S a $\#QMC$ amount of points P. Note that for any measurable set $A \subseteq S$ we have that $\text{Vol}(A) \approx \frac{2bo^d}{|P|} \sum_{p \in P} \chi_A(p)$ ([Asmussen and Glynn, 2007]). We use this method to decrease computation costs.

To ensure that the loss is differentiable, we use a soft inclusion function instead of computing the characteristic function. Given a multibox K, we convert the representation of K to two tensors $Min := \vec{m}_1 \times \ldots \vec{m}_n$ and $Max := \vec{M}_1 \times \ldots \vec{M}_n$. For each point p, we compute

$$\tilde{\chi}_K(p) := \text{relu}(\max_{i \in [n]} \text{mean}(\sigma(\vec{M}_{B_i} - p), \sigma(p - \vec{m}_{B_i})) - 0.5)$$

where σ is the sigmoid function. $\max_{i \in [n]} \text{mean}(\sigma(\vec{M}_{B_i} - p), \sigma(p - \vec{m}_{B_i}))$ is above 0.5 if and only if p is contained in K. Thus we defined the approximated volume function $\text{MonteCarlo}(A) := \frac{2bo^d}{|P|} \sum_{p \in P} \tilde{\chi}_A(p)$

Since we would use models that were trained on $\mathcal{EL}^{++}$ as our benchmark, we design our model only to handle concepts of $\mathcal{EL}^{++}$. This allows us to exploit lemma 1.16 and use normal forms:

- $C \sqsubseteq D$ (NF1).
- $C \sqcap D \sqsubseteq E$ (NF2).
- $C \sqsubseteq \exists R.D$ (NF3).
- $\exists R.C \sqsubseteq D$ (NF4).
- $R_1 \circ R_2 \sqsubseteq S$ (role chain).
- $R \sqsubseteq S$ (role inclusion).

Our loss functions are then:

1. NF1:
$$1 - \frac{\text{MonteCarlo}([\![C]\!] \cap [\![D]\!])}{\text{MonteCarlo}([\![C]\!])}$$

2. NF2:
$$1 - \frac{\text{MonteCarlo}([\![C \sqcap D]\!] \cap [\![E]\!])}{\text{MonteCarlo}([\![C \sqcap D]\!])}$$

3. NF3:
$$1 - \frac{\text{MonteCarlo}([\![C]\!] \cap \pi[[\![R]\!] \cap B(\{-bo\}_{i\in[d]}, \{bo\}_{i\in[d]}) \times [\![D]\!]])}{\text{MonteCarlo}([\![C]\!])}$$

4. NF4:
$$1 - \frac{\text{MonteCarlo}(\pi[[\![R]\!] \cap B(\{-bo\}_{i\in[d]}, \{bo\}_{i\in[d]}) \times [\![C]\!]] \cap [\![D]\!])}{\text{MonteCarlo}(\pi[[\![R]\!] \cap B(\{-bo\}_{i\in[d]}, \{bo\}_{i\in[d]}) \times [\![C]\!]])}$$

5. role chain: Note that for role chain we do not use a fully semantically meaningful loss function, as it would be too large and too computationally expensive to result in useful learning. Instead, we employ a simplified loss function
$$1 - \frac{\text{MonteCarlo}(\pi_0[[\![R_1]\!]] \times \pi_1[[\![R_2]\!]] \cap [\![S]\!])}{\text{MonteCarlo}(\pi_0[[\![R_1]\!]] \times \pi_1[[\![R_2]\!]])}$$

6. role inclusion:
$$1 - \frac{\text{MonteCarlo}([\![R]\!] \cap [\![S]\!])}{\text{MonteCarlo}([\![R]\!])}$$

7.1.2 Evaluation Method

We compare our models over the Hits@K, median reciprocal rank and mean reciprocal rank as implemented in [Kulmanov et al., 2019b]. Given an evaluation dataset $\{C_i \sqsubseteq D_j\}_{i=0}^{n}$ where C_i, D_i are $\mathcal{EL}^{++}$ concepts and a distance function $d : \mathcal{C} \times \mathcal{C} \to \mathbb{R}$, we can generate for each $i \in [n]$ a sequence $\{D_{k_{j,i}}\}_{j=0}^{n}$ such that for any $l_{j,i} \in [n]$ we have that $d(C_i, D_{l_{j,i}}) \leq d(C_i, D_{l_{j+1,i}})$. Denote $\mathscr{R}(i)$ as the rank j of D_i. That is, $D_{l_{\mathscr{R}(i),i}} = D_i$. The Hits@K is then defined as the number of $i \in [n]$ for which $\mathscr{R}(i) < k$. We also measured the mean and median rank of the target concept for each data point. Thus our measurements are:

$$\text{Hit@10} = |\{i \in [n] \mid \mathscr{R}(i) < 10\}| \qquad \text{Hit@100} = |\{i \in [n] \mid \mathscr{R}(i) < 100\}|$$

$$\text{mean} = \frac{1}{n} \sum \mathscr{R}(i) \qquad \text{median} = \text{med}\{\mathscr{R}(i) \mid i \in [n]\}$$

Given a box embedding $[\![\cdot]\!]$, we define the distance of two boxes B_1, B_2 as the norm $d(B_1, B_2) = |\vec{c}_{B_1} - \vec{c}_{B_1}|$. The underlying assumption of the distance score is that the correct subsumptions would be ranked higher. While a volume-based approach for the distance might yield more accurate results, it would be considerably more computationally expensive. Furthermore, it would strain away from the common evaluation metric found in the literature ([Kulmanov et al., 2019b]).

For multibox spaces, given each multibox $[\![C_i]\!] := \bigcup_{j \in J_i} B_{j,i}$ and each box $B_{j,i} = B(\vec{m}_{j,i}, \vec{M}_{j,i})$ we would samples a set of points $P_{j,i}$ from a uniform distribution between the minimal and maximal values $\vec{m}_{j,i}, \vec{M}_{j,i}$. Then we define the measure $d_i(C_i, D_l) = \sum_{p \in P_{j,i}} \chi_{p \in D_l}$.

7.1.3 Choice of Parameters

The choice of parameters for our experiment has proved to be difficult due to the amount of hyperparameters our model used and the lack of time and computational resources required for optimization.

We chose a low dimensional multibox space ($d = 5$) since we observed experimentally that increasing the amount of dimensions resulted in mostly empty intersections between multiboxes and thus no learning over NF2.

We observed that below 10,000 points for the Monte-Carlo algorithm, MultiBoxEL scored only 0 on the evaluation data. We set the amount of points for the Monte-Carlo algorithm to be the highest number we were able to generate without exceeding the allowed memory allocation of our GPU on the lowest amount of boxes per multibox. We ran our experiment with 20,000 sampled points per training session.

We trained each model for 5 epochs as we observed MultiBoxEL plateau on the fifth iteration.

We then set the amount of boxes per multibox to the highest number within our memory allocations (5 boxes per multibox).

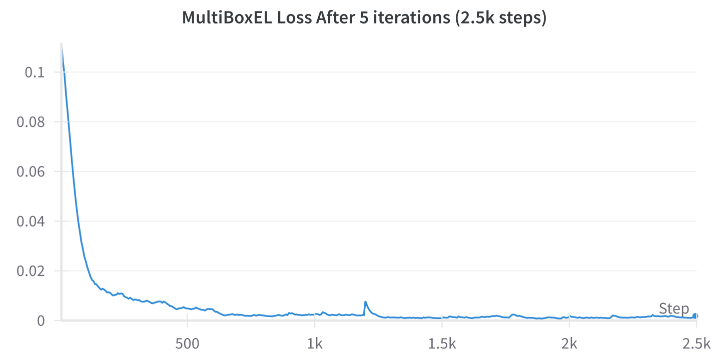


Figure 7.1: MultiboxEL 5 epochs of training

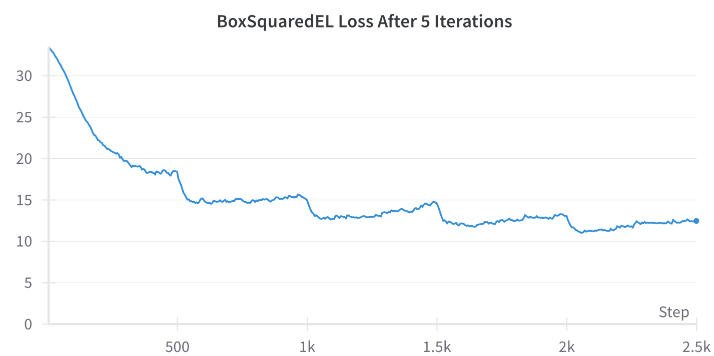


Figure 7.2: BoxSquaredEL 5 epochs of training

In [Jackermeier et al., 2024b] experiment setting, the researchers passed the entire dataset to the model as a tensor. However, this was not possible due to memory limitations. Thus we split the datasets into 5 batches. Under these parameters, the training procedure of MultiBoxEL was close to 8 hours per session.

These hyperparameters represent each concept by a $2 \cdot d \cdot n$ dimensional vector. Hence we evaluated the MultiBoxEL model against BoxEL and BoxSquaredEL with a box space of dimension $d \cdot n$.

It is important to note that although we trained all models for only 5 epochs, BoxEL and BoxSquaredEL did not plateau. Figure 7.2 shows 5 downward jumps in the loss at each epoch (after 500 steps, 1k steps, 1.5k steps, etc.). This is in contract with Figure 7.1 which clearly shows that the training of the multibox squared model quickly plateaus.

7.2 Results

Results

We tested three models: the original box model BoxEL ([Xiong et al., 2022a]), BoxSquaredEL ([Kulmanov et al., 2019b]) and our own model MultiBoxEL on two biomedical ontologies GALEN [Pole, 1996] and Gene Ontology (GO) [Ashburner et al., 2000]. We used the BoxSquaredEL code base as a baseline for our model and as a baseline for the training procedure. For each ontology, our benchmark consists of axioms split into training (80%), validation (10%), and testing (10%) sets for each normal form. Table 7.1 provides us with the number of datapoints in each dataset and the ratios of the normal forms in each dataset.

Dataset	Subset	Role Chain	Role Inclusion	NF1	NF2	NF3	NF4
GALEN	Train	58 (0.09%)	958 (1.42%)	22,299 (33.00%)	10,876 (16.10%)	22,494 (33.30%)	10,877 (16.10%)
	Val	0 (0%)	0 (0%)	2,759 (33.53%)	1,329 (16.15%)	2,803 (34.08%)	1,337 (16.24%)
	Test	0 (0%)	0 (0%)	2,756 (33.42%)	1,346 (16.32%)	2,804 (34.00%)	1,341 (16.27%)
GO	Train	6 (0.01%)	3 (0.00%)	68,376 (65.71%)	9,704 (9.32%)	16,259 (15.62%)	9,703 (9.32%)
	Val	-	-	7,604 (63.77%)	1,177 (9.87%)	1,968 (16.51%)	1,176 (9.86%)
	Test	-	-	7,601 (63.63%)	1,178 (9.86%)	1,987 (16.63%)	1,181 (9.88%)

Table 7.1: Counts and percentages of points in each subset for GALEN and GO datasets

Even though BoxSquaredEL was also originally tested against ANATOMY, we omitted that dataset due to the fact that evaluation over it was too computationally expensive for our available resources. We documented the results in Tables 7.2, 7.3 and 7.4.

	GO MultiBoxEL					GALEN MultiBoxEL				
	NF1	NF2	NF3	NF4	Combined	NF1	NF2	NF3	NF4	Combined
Mean	17886	16742.8	41343.4	19582	22294.4	12238.6	12538.2	27690.6	13326.2	16790.4
Median	15725	12912.2	41344.4	18819	22033.2	11490.75	**11191.75**	23142	13653.5	19168
Top10	**0.06**	0	0	0	**0.028**	0.018	0	0	0	0.012
Top100	**0.068**	0	0	0	**0.032**	0.042	0	0.002	0.004	0.018

Table 7.2: Results for GO MultiBoxEL and GALEN MultiBoxEL with best scoring model on each column bolden

	GO BoxSquaredEL					GALEN BoxSquaredEL				
	NF1	NF2	NF3	NF4	Combined	NF1	NF2	NF3	NF4	Combined
Mean	**9740.8**	**16025.8**	23171.6	20476.8	**13636.4**	**5381.4**	**11928**	**12116.4**	**10918.6**	**9633**
Median	**5402.6**	**14871.4**	23864.8	19795.4	10145.2	3480.4	12206.6	11540.8	**10567.8**	**8797.2**
Top10	0.02	0.002	0	0	0.006	0.004	0	0	0	0.002
Top100	0.056	**0.02**	0.002	0	0.04	0.054	0.008	0.048	0	**0.032**

Table 7.3: Results for GO BoxSquaredEL and GALEN BoxSquaredEL with best scoring model on each column bolden

	GO BoxEL					GALEN BoxEL				
	NF1	NF2	NF3	NF4	Combined	NF1	NF2	NF3	NF4	Combined
Mean	13857	20360.8	**20744**	**14589.8**	15707.4	6482.2	14362.4	10643.4	10320.8	9796.2
Median	5854.4	19885.4	**20520.8**	**8204**	**9780.4**	**1891.2**	15609.4	**10449.6**	11081	9517.2
Top10	0.01	0	0	**0.476**	0.05	**0.046**	0	0	**0.11**	**0.036**
Top100	0.046	0	0	**0.476**	0.082	**0.198**	0	**0.02**	**0.266**	**0.118**

Table 7.4: Results for GO BoxEL and GALEN BoxEL with best scoring model on each column bolden

7.2.1 Discussion

Upon analysis, the multibox approach appears to outperform the other models in the Hits@K GO tasks and the GALEN NF2 median rank. Figure 7.3 shows that MultiBoxEL learned a good representation quickly on NF1. However, for the other normal forms, the loss function of MultiBoxEL was not as efficient at finding a gradient towards a direction of descent, as can be seen in Figures 7.5 and 7.6. A similar analysis of the loss function over GALEN reveals that this pattern is consistent over the training of the MultiBoxEL model.

These results suggest that MultiBoxEL performed well on GO specifically due to the high ratio of NF1 data points in the dataset. Recall that the proof of incompleteness result of box spaces (Theorem 4.37) and the completeness of multibox spaces (Theorem 5.20) relied solely on subsumption relations of type NF1 and NF2. This might suggest that the reason for the high success of MultiBoxEL in those tests is due to the expressivity of the representation over NF1.

Figures 7.3 through 7.6 compare the three models'—Boxsqel (blue), Boxel (orange), and our model, MultiboxEL(green)—on the GO dataset. The vertical axis represents the normalized loss values, scaled between the loss function values of all three models for a common range. The horizontal axis indicates the number of computational steps up from the beginning of training up to the end of last, fifth, epoch

In each training epoch we shuffled the data and divided it to equally sized batches. This shuffling introduces variability in the batches the model sees, which can impact the loss values between steps.

In Figure 7.4, the BoxSquaredEL model (blue line) demonstrates the behavior we expect from a learning algorithm under such conditions. Initially, the loss decreases sharply, reflecting a rapid reduction in error as the model begins to learn from the data. This steep drop indicates effective early-stage learning.

However, at the beginning of a new epoch, the model's loss exhibits oscillations. These fluctuations occur because the model encounters slightly different combinations of data in each batch, leading to variability in the optimization path. Despite these oscillations, BoxSquaredEL continues to show an overall downward trend, confirming that it is learning and making progress. In Figures 7.3 and 7.5, we barely see any learning of BoxSquaredEL, which reflects a poor learning procedure. However, BoxSquaredEL still maintained a competitive edge in our evaluation.

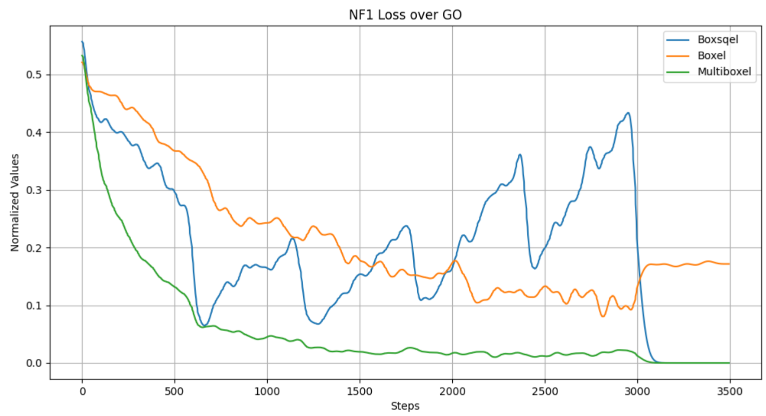


Figure 7.3: NF1 Loss over GO

In Figures 7.5 and 7.3 we see that MultiBoxEL, learns a representation early on and then plateaus for the other 4 epochs while in 7.4 and 7.6 we see an immediate plateau.

The early plateauing behavior of MultiBoxEL indeed suggests that the model might be falling into local minima, unable to escape and continue improving its performance. This kind of behavior often arises when the optimization process gets stuck at a suboptimal point in the loss landscape, rather than converging to a global minimum. Adding regularizing factors can help address this issue by encouraging the model to explore the loss landscape more thoroughly and avoid overfitting to these local minima.

We did not expect the multibox model to perform poorly on the NF3 and NF4 tasks since the multiboxes model uses the most semantically accurate representation of the existential operator out of the three models. Out of the three models, BoxEL consistently performed the best on NF3 and NF4 despite having the least expressive representation for relations. Both BoxSquaredEL and MultiBoxEL represent relations using two boxes, one for the domain and one for the codomain of the relation (alternatively, as a product of two boxes and a projection to the embedding space). Due to the complexity of the expression for the existential operator, it is possible that the loss function for NF3 and NF4 is too complicated for the algorithm to find a local minimum in the given number of epochs. [Jackermeier et al., 2024b] claims that BoxSquaredEL performs better than BoxEL on NF3. The failure in replication could be the result of running the experiments on low dimensional spaces and for less time.

To test this hypothesis, we trained both BoxSquaredEL and MultiBoxEL on a smaller dataset, for which we could run 300 epochs. We saw that BoxSquaredEL managed to converge to a good representation while MultiBoxEL could not.

Furthermore, we can see in 7.2.1 that MultiBoxEL achieves a lower loss by

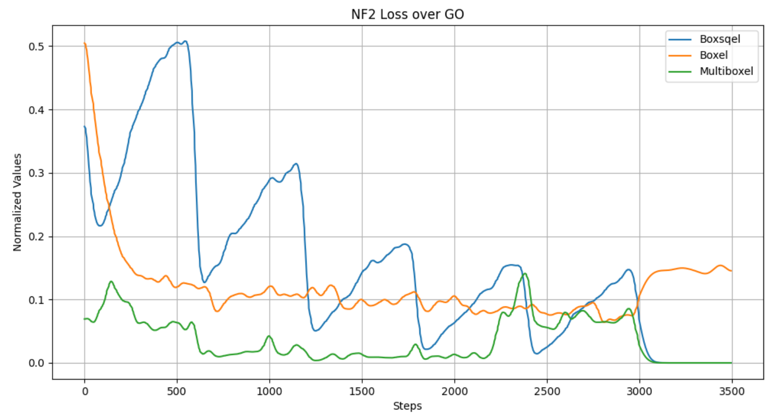


Figure 7.4: NF2 Loss over GO

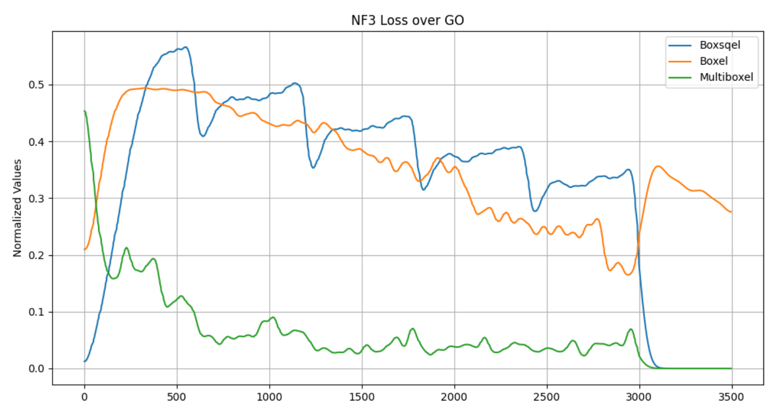


Figure 7.5: NF3 Loss over GO

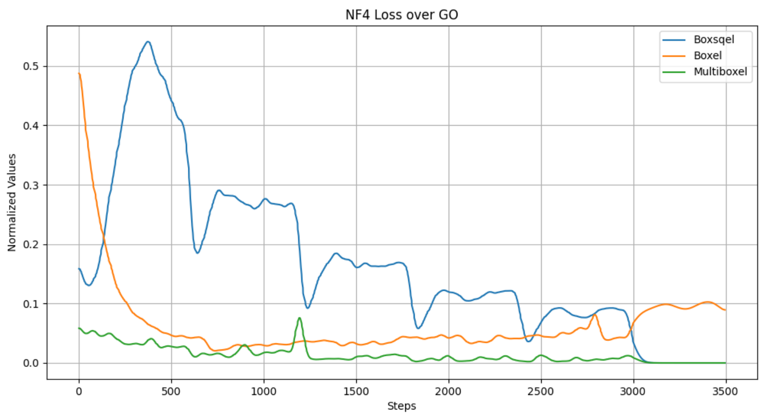


Figure 7.6: NF4 Loss over GO

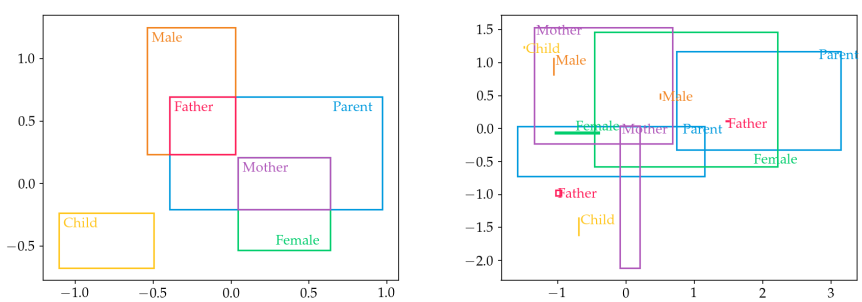


Figure 7.7: BoxSquaredEL (left) and MultiBoxEL (right) representation of a family dataset after 300 epochs

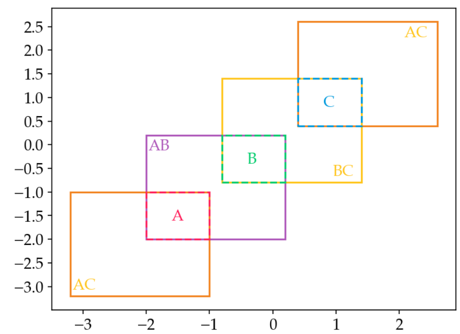


Figure 7.8: Caption

minimizing the areas of the boxes instead of attempting to satisfy the subsumption relations. This could have been avoided by adding regularization factors to MultiBoxEL to prevent such learning.

In Figure 7.8, we see a visualization of a multibox representation of the knowledge base from the incompleteness result in Theorem 4.37. In Figure 7.2.1, we see a visualization of embedding spaces, based on the same knowledge base as training data, generated by the BoxSquaredEL and MultiBoxEL models. As expected, neither of them correctly represented the knowledge base.

In the BoxSquaredEL representation, concept C is not subsumed by concept AC and BC, concept $AB \sqcap AC$ is not subsumed by A, concept B is not subsumed by AB and is incorrectly intersecting concept AC. An advantage of the BoxSquaredEL representation is that the boxes area remains large (except for concept C) which helps in interpretability.

In the MultiBoxEL representation, most of the concepts are not correctly subsumed. The only pair that is correctly related to one another is C and AC. It appears that the model has minimized the loss function by reducing the areas of the subsumed concepts (see concept A). This further shows the need for regularization and the tendency of MultiBoxEL towards weak local minima.

7.3 Summary

This chapter presents an experimental evaluation of three models for knowledge representation: the original box model BoxEL, the state of the art BoxSquaredEL model, and a new multibox-based model MultiBoxEL. The experiments were conducted on two biomedical ontologies: GALEN and Gene Ontology (GO), with a focus on the models' performance on different normal forms (NF1, NF2, NF3, NF4).

The computational complexity of the MultiBoxEL made the choice of parameters and optimization difficult. Training a multibox model is only feasible

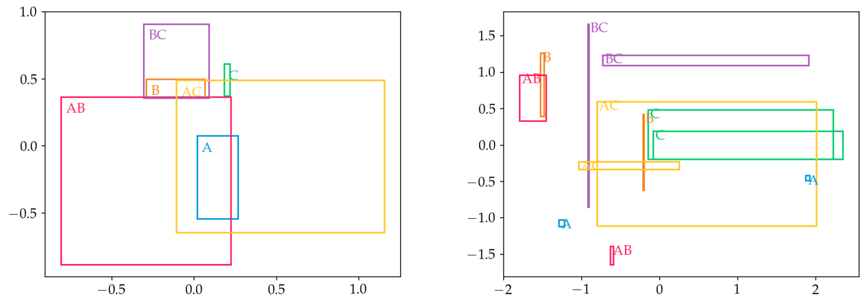


Figure 7.9: BoxSquaredEL (left) and MultiBoxEL (right) representation of a incompleteness result dataset after 300 epochs

within a low-dimensional vector space due to the high memory and computational requirements of handling higher-dimensional spaces. As the number of dimensions increases, the model's training becomes more computationally expensive, and the resulting intersections between multiboxes often become sparse, leading to inefficiencies in learning. Therefore, a low-dimensional multibox space (with $d = 5$) was selected to balance expressivity and computational feasibility, but this choice limits the expressivity of the model.

To compute the multibox loss function effectively, we had to rely on a Monte Carlo method for sampling points due to the complexity of the multiboxes and the high computational demands of calculating their intersections. We used 20,000 sampled points per training session, which was the maximum number we could process without exceeding the GPU's memory allocation. This choice of Monte Carlo sampling allowed us to approximate the loss function over the multiboxes, but also introduced challenges in ensuring that the samples were sufficiently representative while staying within memory constraints. The performance of the model was highly dependent on the number of samples, and insufficient sampling could result in poor learning or failure to compute meaningful gradients for optimization.

In the results, MultiBoxEL outperformed the other models on the Hits@K GO tasks and the GALEN NF2 median rank, likely due to the high proportion of NF1 data points in the datasets. However, it struggled with other normal forms (NF3 and NF4), where BoxEL performed better despite having a less expressive representation. This failure was attributed to the complexity of the NF3 and NF4 tasks and the difficulty of the loss function in finding a suitable minimum within the limited number of epochs. Additionally, the need for regularization became apparent, as the MultiBoxEL model exhibited a tendency to fall into weak local minima. The absence of regularization resulted in suboptimal training, with the model minimizing the box areas rather than satisfying the necessary subsumption relations. This further emphasizes the importance of adding regularizing factors to help the model explore the loss landscape more effectively and avoid overfitting to poor minima.

Part IV

Conclusion

In this thesis, we set up a strong ground mathematical foundation for working with box space embeddings and offered potential extensions of the system for complete representation of knowledge bases.

We proved important results for $\mathcal{EL}^{++}$, including its incompleteness against box space embeddings and the fact that every satisfiable $\mathcal{EL}^{++}$ knowledge base has a finite model.

We motivated the usage of hyperdoctriens for the research of knowledge base embeddings by demonstrating how they can be used to discover the left and right adjoints and thus the existential and universal operators for different embeddings and used hyperdoctrines to extend box spaces to spaces for which $\mathcal{EL}^{++}$ and $\mathcal{ALC}$ are complete.

Lastly, we tested our new embeddings and compared the results to other models in the literature. While the results point to the fact that there are a class of datasets that could not currently be represented using box space models, further work needs to be done before we can conclude whether a multibox approach is a practical alternative. Primarily, the data analysis shows that the loss function of multibox spaces needs to be revised - regularized to avoid boxes, simplified to avoid vanishing gradients and to be made more differentiable and lastly, further optimizations should be explored in order to decrease the run time required for learning.

Furthermore, we refrained from testing our multibox model against $\mathcal{ALC}$. Further work would be needed to verify the expressibility and computational requirements of a multibox model against an $\mathcal{ALC}$ dataset.

Nevertheless, our analysis demonstrates that even in the current state, multibox spaces power of expressibility could make them more useful for handling datasets enriched with subsumptions relations of type NF1 and NF2.

Bibliography

[Aharoni, 2024] Aharoni, A. (2024). Pushingenvelope.

[Ashburner et al., 2000] Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., and Sherlock, G. (2000). Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nat Genet*, 25(1):25–29.

[Asmussen and Glynn, 2007] Asmussen, S. and Glynn, P. (2007). *Stochastic Simulation: Algorithms and Analysis*. Stochastic Modelling and Applied Probability. Springer New York.

[Avron et al., 2008] Avron, A., Dershowitz, N., and Rabinovich, A. (2008). *Boris A. Trakhtenbrot: Academic Genealogy and Publications*, pages 46–57. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Baader, 2003] Baader, F. (2003). Terminological cycles in a description logic with existential restrictions. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, page 325–330, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Baader et al., 2005] Baader, F., Brandt, S., and Lutz, C. (2005). Pushing the el envelope. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI'05, page 364–369, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Baader et al., 2017] Baader, F., Horrocks, I., Lutz, C., and Sattler, U. (2017). *An Introduction to Description Logic*. Cambridge University Press.

[Birkhoff, 1967] Birkhoff, G. (1967). *Lattice Theory*. American Mathematical Society, Providence, 3rd edition.

[Bordes et al., 2013] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

[Brandt, 2004] Brandt, S. (2004). Polynomial time reasoning in a description logic with existential restrictions, gci axioms, and - what else? In *European Conference on Artificial Intelligence*.

[Gutiérrez-Basulto and Schockaert, 2018] Gutiérrez-Basulto, V. and Schockaert, S. (2018). From knowledge graph embedding to ontology embedding? an analysis of the compatibility between vector space representations and rules.

[Haykin, 1999] Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. Prentice Hall.

[Jackermeier et al., 2024a] Jackermeier, M., Chen, J., and Horrocks, I. (2024a). Dual box embeddings for the description logic el++.

[Jackermeier et al., 2024b] Jackermeier, M., Chen, J., and Horrocks, I. (2024b). Dual box embeddings for the description logic el++.

[Klee, 1977] Klee, V. (1977). Can the measure of n 1 [a i , b i] be computed in less than o(n logn) steps? *American Mathematical Monthly*, 84:284.

[Kulmanov et al., 2019a] Kulmanov, M., Liu-Wei, W., Yan, Y., and Hoehndorf, R. (2019a). El embeddings: Geometric construction of models for the description logic el ++.

[Kulmanov et al., 2019b] Kulmanov, M., Liu-Wei, W., Yan, Y., and Hoehndorf, R. (2019b). El embeddings: Geometric construction of models for the description logic el ++. *ArXiv*, abs/1902.10499.

[Lawvere, 1969] Lawvere, F. W. (1969). Adjointness in foundations. *Dialectica*, 23:281–296.

[Nguyen et al., 2021] Nguyen, T., Raghu, M., and Kornblith, S. (2021). Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth.

[Overmars and Yap, 1991] Overmars, M. H. and Yap, C.-K. (1991). New upper bounds in klee's measure problem. *SIAM Journal on Computing*, 20(6):1034–1045.

[Pole, 1996] Pole, R. P. (1996). The galen high level ontology.

[Rector and Horrocks, 1997] Rector, A. L. and Horrocks, I. (1997). Experience building a large, re-usable medical ontology using a description logic with transitivity and concept inclusions.

[Riehl, 2017] Riehl, E. (2017). *Category theory in context*. Aurora: Dover modern math originals. Dover Publications.

[Schmidt-Schauß and Smolka, 1991] Schmidt-Schauß, M. and Smolka, G. (1991). Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26.

[Staab and Studer, 2009] Staab, S. and Studer, R. (2009). *Handbook on Ontologies*. Springer Publishing Company, Incorporated, 2nd edition.

[van Harmelen et al., 2007] van Harmelen, F., van Harmelen, F., Lifschitz, V., and Porter, B. (2007). *Handbook of Knowledge Representation*. Elsevier Science, San Diego, CA, USA.

[Xiong et al., 2022a] Xiong, B., Potyka, N., Tran, T.-K., Nayyeri, M., and Staab, S. (2022a). Faithiful embeddings for el++ knowledge bases.

[Xiong et al., 2022b] Xiong, B., Potyka, N., Tran, T.-K., Nayyeri, M., and Staab, S. (2022b). Faithiful embeddings for el++ knowledge bases.