Universidade Estadual de Campinas
Instituto de Filosofia e Ciências Humanas

Gabriel Rodrigues Rocha

Towards Ontological Heterogeneity: Splicing Logic and Ontology

Em Direção à Heterogeneidade Ontológica: Tecendo Lógica e Ontologia

Campinas
2024

Gabriel Rodrigues Rocha

Towards Ontological Heterogeneity: Splicing Logic and Ontology

Em Direção à Heterogeneidade Ontológica: Tecendo Lógica e Ontologia

Dissertação apresentada ao Instituto de Filosofia e Ciências Humanas da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Filosofia.

Dissertation presented to the Institute of Philosophy and Human Sciences of the University of Campinas in partial fulfillment of the requirements for the degree of Master, in the area of Philosophy.

Orientador: Walter Alexandre Carnielli

ESTE TRABALHO CORRESPONDE À VERSÃO FINAL DA DISSERTAÇÃO DEFENDIDA PELO ALUNO GABRIEL RODRIGUES ROCHA, E ORIENTADO PELO PROF. DR. WALTER ALEXANDRE CARNIELLI.

Campinas
2024

Ficha catalográfica
Universidade Estadual de Campinas (UNICAMP)
Biblioteca do Instituto de Filosofia e Ciências Humanas
Cecília Maria Jorge Nicolau - CRB 8/3387

R582e Rocha, Gabriel Rodrigues, 1997-
Em direção à heterogeneidade ontológica : tecendo lógica e ontologia / Gabriel Rodrigues Rocha. – Campinas, SP : [s.n.], 2024.

Orientador: Walter Alexandre Carnielli.
Dissertação (mestrado) – Universidade Estadual de Campinas (UNICAMP), Instituto de Filosofia e Ciências Humanas.

1. Lógica. 2. Ontologias (Recuperação da informação). 3. Heterogeneidade. I. Carnielli, Walter Alexandre, 1952-. II. Universidade Estadual de Campinas (UNICAMP). Instituto de Filosofia e Ciências Humanas. III. Título.

Informações Complementares

Título em outro idioma: Towards ontological heterogeneity : splicing logic and ontology
Palavras-chave em inglês:
Logic
Ontologies (Information retrieval)
Heterogeneity
Área de concentração: Filosofia
Titulação: Mestre em Filosofia
Banca examinadora:
Walter Alexandre Carnielli [Orientador]
Abílio Azambuja Rodrigues Filho
Julio Cesar dos Reis
Data de defesa: 26-08-2024
Programa de Pós-Graduação: Filosofia

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0009-0003-9687-3921
- Currículo Lattes do autor: http://lattes.cnpq.br/4375411806886245

UNICAMP

Universidade Estadual de Campinas
Instituto de Filosofia e Ciências Humanas

A Comissão Julgadora dos trabalhos de Defesa de Dissertação de Mestrado composta pelos Professores Doutores a seguir descritos, em sessão pública realizada em 26 de agosto de 2024, considerou o candidato Gabriel Rodrigues Rocha aprovado.

Prof. Dr. Walter Alexandre Carnielli

Prof. Dr. Abílio Azambuja Rodrigues Filho

Prof. Dr. Julio Cesar dos Reis

A Ata de Defesa com as respectivas assinaturas dos membros encontra-se no SIGA/Sistema de Fluxo de Dissertações e na Coordenadoria do Programa de Pós-Graduação em Filosofia do Instituto de Filosofia e Ciências Humanas.

To my beloved Sofia

Acknowledgments

No piece of work is the result of a single pair of hands. There is a certain poetry to writing acknowledgements, and although I lack the skills of a wordsmith, this is my attempt at a small homage to the extra pairs of hands that have contributed. As it stands, this is perhaps the most important, if not the only, section of this work.

Sofia, as cliché as it may read, I could not have done it without you. Thank you for your unending support and love, for enduring more than a few lonely nights as I spent dusk to dawn writing, for your compassion and for not losing interest despite all my nonsensical blabbering. Thank you for tolerating me and, most of all, thank you for being the only thing that matters.

To my mother, Monica, and my father, Joel, I could not have gotten close to starting this work without the education and support you have provided me. Thank you for everything and for steering me into the path of education. Thank you to my brother, Jean, for all of our conversations and laughs. Hopefully I get an acknowledgement note when you finish your future thesis as well.

To my supervisor, Prof. Dr. Walter Carnielli, I am grateful for your receptiveness, assertiveness, kindness, friendliness and, most of all, for your patience! Your role was instrumental in helping me become a better researcher. To my former supervisor and, I must say, "cosupervisor", Prof. Dr. José Guimarães, for convincing me to start this journey in the first place — thank you for everything! A special thanks to other professors and researchers that had an impact on this work: Marcelo Coniglio, Júlio dos Reis, Abilio Rodrigues, Ana Cristina Sernadas, Fabio Bertato, Branden Fitelson, Pierre Grenon, Till Mossakowski, Fabian Neuhaus, and many others. I specially thank the researchers at the *Laboratory for Applied Ontology* (LOA) of the *Institute for Cognitive Sciences and Technologies* (ISTC), particularly Nicola Guarino and Stefano Borgo for not only working on foundational formal ontology and logics for knowledge representation, but also for sparking some of the ideas in my work.

Thank you to my fellow classmates and colleagues: Rafael Ongaratto, Deiver Melo, Alan René, Mahan Vaz, Sofia Meirelles, Renato Leme, Daniel Diniz, Daniel Gieseler, Maíra Bittencourt, Mariana Fernandes, Raoni Arroyo. We may not have spoken much, but I deeply appreciate all of our conversations and all of your effort to keep the research community alive.

I deeply appreciate and thank all of my friends: André, Nick, 3M, Sabino, Bona, Leo, Breno, Jajá, Lelê, Bilio. Thanks for all the moments we shared and all future moments we will share. Special thanks to Rapha, for empathizing with the arduous task of writing a thesis and for contributing to science, and to Bertanha, for proofreading early drafts and commenting on my work. Thank you to all my teammates at work, who I am happy to call friends, for making my job lighter than it could have been. I specially thank Mihail, for the support, understanding and trust in me.

Thank you to the Federal University of São Carlos (UFSCar) for being my *alma mater*, and for allowing me to organize activities and projects. Thank you, Hackoonspace team, for keeping these activities alive. To Prof. Dr. Gustavo Vieira, I must state my gratitude for that one beer we had, and to Prof. Dr. Rodrigo Rodrigues, for believing in my jazz project. Thanks Alexandra Elbakyan, for Sci-Hub, and to the anonymous creators of Library Genesis. Thanks for keeping science open.

I thank the State University of Campinas (Unicamp), for providing public and high-quality education, and for being a platform upon which I can stand. Lastly, I thank the reader, for a text with no readers is similar to a falling tree with no listeners: does it even make a sound?

My back to the wall
A victim of laughing chance
This is for me
The essence of true romance
Sharing the things we know and
love with those of my kind
Libations, sensations
That stagger the mind

— Steely Dan, *Deacon Blues*

Resumo

Lógica e Ontologia são áreas da Filosofia unidas como gêmeas em termos de suas origens, mas separadas por suas motivações e abordagens. A Ontologia Aplicada é um ramo da Ontologia que descende da prática analítica iniciada por Frege e Russell e que se propõe a fornecer métodos, ferramentas e conceitos para se praticar Ontologia de modo pragmático, com um viés computacional e que muito se relaciona com Lógica. Esta dissertação tem como objetivos clarificar a relação entre Ontologia e Lógica, ao expor quais são e como são representados os objetos de estudo da Ontologia Aplicada; e tratar de possíveis abordagens para obter o que é chamado de heterogeneidade ontológica. A heterogeneidade ontológica está relacionada a um problema metateórico fundamental: o que é uma ontologia "correta" e quantas existem? Ao assumir uma resposta pluralista para essa pergunta, é necessário propor métodos e técnicas que permitam processos de refinamento, integração, conexão e decomposição de ontologias — esse conjunto de processos é o que constitui, de fato, a heterogeneidade ontológica. Inspirando-se na proposta de heterogeneidade ontológica de Mossakowski et. al, chamada Carnapiana-Goguenista, a dissertação propõe uma nova abordagem chamada de da Costa-Tarskiana — embasada na visão de pluralismo lógico defendida por da Costa e em operadores de consequência de Tarski. Por fim, a dissertação trata de uma classe de lógicas comumente utilizada para descrever ontologias, as chamadas lógicas de descrição, e alguns de seus problemas em aberto e implicações metodológicas que podem ser formalizadas e tratadas na abordagem da Costa-Tarskiana.

Palavras-chave: Lógica; Ontologias; Heterogeneidade.

Abstract

Logic and Ontology are regarded as twins in Philosophy due to their origins, yet separated by their motivations and approaches. Applied Ontology is a branch of Ontology which descends from the analytical view as touted by Frege and Russel, and whose goal is to provide methods, tools and concepts to practice Ontology in a pragmatic manner, with a computational bias linked to Logic. This dissertation aims to clarify the relationship between Ontology and Logic, by exposing what are the objects of Applied Ontology and how they are represented; and also to present possible ways to achieve what is termed ontological heterogeneity. Ontological heterogeneity is closely linked to a fundamental metatheoretical problem: what is a "correct" ontology and how many are there? By assuming a pluralist answer to this question, it is necessary to propose methods and tools to refine, integrate, connect and decompose ontologies — indeed, this set of processes constitutes what is constitutes ontological htereogeneity. Drawing from the heterogeneous proposal due to Mossakowski et. al, termed Carnapian-Goguenism, this dissertation proposes a novel approach termed da Costian-Tarskianism, based on da Costa's view on logical pluralism and Tarski's consequence operators. Lastly, this dissertation discusses a class of logics which is widely used to represent ontologies, so-called description logics, and some of their open problems and methodological implications that may be formalized and handled by the da Costian-Tarskianist approach.

Keywords: Logic; Ontologies; Heterogeneity.

List of Figures

List of Tables

List of Abbreviations and Acronyms

ABox Assertional Box. 113

AI Artificial Intelligence. 32

AMOP Applied Meta-Ontological Problem. 20, 63, 128

BFO Basic Formal Ontology. 19, 22, 32, 46, 127, 164

CL Common Logic. 47

CLIF Common Logic Interchange Format. 19, 47, 107, 115

CoFI Common Framework Initiative. 77

DAML+OIL DARPA Agent Markup Language + Ontology Inference Layer. 53, 107

DDL Distributed Description Logics. 89

DL Description Logic. 22, 109

DOL Distributed Ontology, Model and Specification Language. 77, 90

DOLCE Descriptive Ontology for Linguistic and Cognitive Engineering. 19, 51, 127

ESDG Extended Splittable Development Graph. 105

FMA Foundational Model of Anatomy. 60

FOL First-Order Logic. 111

GCI General Concept Inclusion. 113

GFO General Formal Ontology. 58

GO Gene Ontology. 59

GUM Generalized Upper Model. 58

Hets Heterogeneous Toolset. 21, 89

IOF Industrial Ontologies Foundry. 43

KIF Knowledge Interchange Format. 19, 53, 107

KR Knowledge Representation. 32

LFI Logic of Formal Inconsistency. 118

LLM Large Language Model. 129

MILO Mid-Level Ontology. 54

MLP Meta-Logical Problem. 20, 64

MOP Meta-Ontological Problem. 62, 127

NEMO Ontology and Conceptual Modeling Research Group. 58

NIST US National Institute of Standards and Technology. 130

OBDA Ontology-based Data Access. 19, 22

OBO Open Biological and Medical Ontologies. 43

OBOF Open Biomedical Ontologies Format. 107

OCHRE Object-Centered High-Level Reference Ontology. 58

OIL Ontology Inference Layer. 115

OMS Ontology, Model and Specification [system]. 90

OntoCommons Ontology-Driven Data Documentation for Industry Commons. 43

OWL The Web Ontology Language. 19, 22, 47, 107, 111

PNC Principle of Non-Contradiction. 91

RDF Resource Description Framework. 19, 53

SUMO Suggested Upper-Merged Ontology. 19, 54

TBox Terminological Box. 113

TPTP Thousands of Problems for Theorem Provers. 90

UFO Unified Foundational Ontology. 19, 56, 127

UML Unified Modeling Language. 57

W3C World Wide Web Consortium. 111

YAMATO Yet Another More Advanced Top-level Ontology. 58

Contents

Chapter 1

Introduction

THIS is an essay on the interconnections between various domains of knowledge, from several points of view. It examines the interconnections between past and present, theory and practice, Ontology and Logic, Computer Science and Philosophy. The unifying element among these domains lies in a fundamental problem termed the meta-ontological problem. In informal terms, the meta-ontological problem may be understood as the questions of what it means for a conceptualization to be correct and whether there are any correct conceptualizations. This key problem serves as the foundation upon which the essay is constructed. The following is a summarized overview of the contents of each chapter.

Chapter 2, entitled "*Ontology and ontologies*", discusses the history of Ontology, the science of what there *is*, and how its sub-branch of applied Ontology came to be, influenced by computer science, knowledge representation and information management. The chapter offers an explanation of the ways in which applied Ontology shares key characteristics of Ontology as a field, and it presents a historical examination of the philosophy of applied Ontology. The chapter's goal is to elucidate how Ontology started as a synonym to philophical inquiry, and how it came to be a term discussed in computer science and artificial intelligence, such as in the works of McCarthy and P. Hayes [251].

Additionally, chapter 2 delineates a clear distinction between Ontology as a field of study and ontologies as the subjects of study. In the context of this essay, an **o**ntology, with boldface O, is always regarded as an object within the field of applied Ontology. The chapter presents and discusses Gruber's widely popular definition of what is an ontology, as reproduced below in definition 1.0.1.

Definition 1.0.1 (Ontology by Gruber)**.** An *ontology* is an explicit specification of a conceptualization.

However, Gruber's definition is not sufficiently formal for the purposes of concretely constructing an **o**ntology. Chapter 2 presents Guarino's widely acknowledged definition of what is an **o**ntology, a formalized account of Gruber's definition loosely inspired by W. V. O. Quine's ontological commitments. This essay presents three formal definitions for **o**ntology, the first of which is reproduced in definition 1.0.2 below.

Definition 1.0.2 (Ontology)**.** Let $\mathbf{C}$ be a conceptualization, $\mathbf{L}$ be a logical language with vocabulary $\mathbf{V}$ and $\mathbf{K}$ be an ontological commitment. An ontology $\mathbf{O_K}$ for $\mathbf{C}$ with vocabulary $\mathbf{V}$ and ontological commitment $\mathbf{K}$ is a logical theory consisting of a set of formulas **L** designed so that the set of its models approximate as well as possible $\mathbf{I_K}(\mathbf{L})$, the set of intended models of **L** according to **K**.

Lastly, chapter 2 briefly discusses about the philosophy of applied Ontology and how it relates to how the practice of applied Ontology. In general, it may be argued that the practice of applied Ontology is quite pragmatic, however it is not devoid of philosophical concerns.

Chapter 3, entitled "*From Metaphysics to Industry*", links the theory of applied Ontology with its practical applications. The chapter presents the diversity of **o**ntologies in the literature, by discussing a number of examples of ontologies (e.g. BFO, DOLCE, SUMO, UFO), their representation languages (e.g. OWL, CLIF, KIF, RDF), and the philosophical choices that inform them. The chapter focuses on what are called top-level or foundational **o**ntologies, **o**ntologies whose goal is to represent very general concepts, independent of specific domains. Top-level **o**ntologies are frequently explicit about their philosophical choices, and attempt to serve as the basis or foundation for other **o**ntologies.

Additionally, chapter 3 also presents an extension of Guarino's taxonomy of **o**ntologies pictured in figure 1 below. To demonstrate the practical aspect of **o**ntologies, the chapter offers an examination of domain **o**ntologies utilized by researchers and industry professionals on a daily basis, in domains such as security, engineering, management, oil business, food industry, biology, among others.

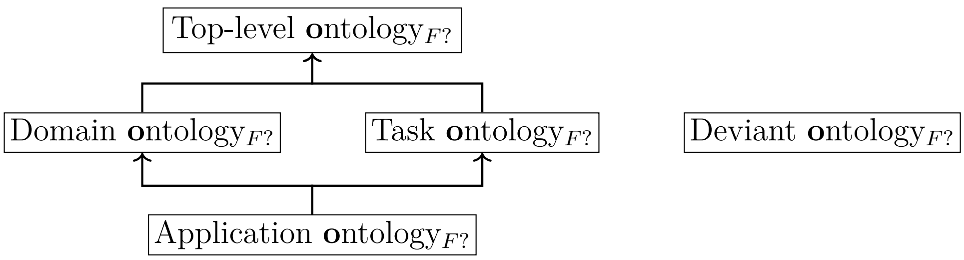


Figure 1.1: An extension of Guarino's taxonomy of ontologies. The subscript $F?$ denotes an **o**ntology optionally belongs to a foundry.

It is noteworthy that there exists high ontological diversity in all facets of what constitutes an **o**ntology, and to reason over more than a single **o**ntology is often desirable or necessary. This relates to the concepts of *ontological heterogeneity* and *interoperability* [392], which are fundamental to this essay. Ontological heterogeneity may be defined as a scenario in which two **o**ntologies make different ontological assumptions about the same domain, potentially using different representation languages. Furthermore, ontological interoperability refers to the ability to reason over **o**ntologies in a heterogeneous scenario. Heterogeneity and interoperability are not limited to computational systems, such as Ontology-Based Data Access (OBDA) systems, however the concepts are frequently seen as causes for complexity in this context. Introducing heterogeneity may difficult interoperability, thereby introducing the need to develop methods and tools. Chapter 3 discusses some of the tooling that exists in the literature for computer systems. The matter of developing a set of methods for interoperability is discussed extensively in the domain of formal representation in chapters 7 and 6.

Chapter 4, entitled "*The Meta-Ontological Problem*" discusses the aforementioned meta-ontological problem, hence its namesake. The chapter formally defines the problem and presents arguments for a reducing the problem from the ontological level into the logical level, if one restricts the problem to the field of applied Ontology.

Definition 1.0.3 (Applied Meta-Ontological Problem)**.** The Applied Meta-Ontological Problem (AMOP) may be informally defined as the following two-part questions:

1. What does it mean for an **o**ntology to be *correct*?
2. How many correct **o**ntologies are there if any?

The chapter's arguments lead to the formulation of a reduction thesis which is based on observations from chapter 3. The reduction thesis postulates that ontologies depend on logics in the context of applied Ontology. Consequently, the problem of correctness of an ontology may be reduced to the problem of correctness of a logic, termed the meta-logical problem.

Definition 1.0.4 (Meta-Logical Problem)**.** The Meta-Logical Problem (MLP) may be informally defined as the following two-part questions:

1. What does it mean for a logic to be *correct*?
2. How many correct logics are there if any?

Thesis 1.1 (AMOP-MLP Reduction Thesis)**.** The AMOP is partially reducible to the MLP via a linguistic argument. In other words, solving the MLP is one of the steps required to solve the AMOP.

There are two main, contrasting ways to address the meta-logical problem: logical monism and logical pluralism. Chapter 4 presents arguments for the reduction thesis and for logical pluralism based on pragmatic and empirical observations, thereby placing this essay within the broader context of logical pluralism.

One consequence of adopting the reduction thesis is that one must select a particular view of logical pluralism upon which to base an approach towards ontological pluralism. Chapter 5, entitled "*From Ontology to Logic*", addresses this matter by examining the various views of logical pluralism. The chapter offers a historical overview of different forms of logical pluralism, demonstrating the pervasiveness of this concept in contemporary scientific thought. Among the various forms of pluralism, two stand out by exhibiting a sense of duality — *Carnapian pluralism* and *da Costa pluralism*. A key point of chapter 5 is that logical or ontological pluralism are not sufficient to achieve ontological heterogeneity and interoperability. Chapter 5 examines the heterogeneous framework developed by Kutz and Mossakowski [212] based on Carnapian pluralism, named *Carnapian-Goguenism*.

Carnapian-Goguenism redefines the concept of what is an **o**ntology by using category-theoretical constructs from institution theory, developed by Joseph Goguen, and graph theory. In the Carnapian-Goguenist framework, an **o**ntology is defined with respect to a development graph (also called a *hyperontology*).

Definition 1.0.5 (**O**ntologies and hyperontologies)**.** An *abstract structured heterogeneous ontology* with respect to some logic graph, viz. an **o**ntology, is a node O in a development graph $\mathcal{DG}$ whose underlying institution is a Grothendieck institution built over the same logic graph. A *hyperontology* is the entire development graph.

Specific operations over a development graph are what allow three different possibilies of relating **o**ntologies. These possibilities are what ensure that Carnapian-Goguenism is a fully heterogeneous framework which supports interoperability at the theoretical level. The possibilities are:

- *Refinement*: an **o**ntology can be refined into another by specifying a mapping which translates the former into the latter.
- *Integration*: two **o**ntologies can be mapped into a third *existing* reference ontology.
- *Connection*: two **o**ntologies can be related via some additional interface **o**ntology, usually specified manually, which is used to *generate* an overall third **o**ntology.

Chapter 5 extensively discusses each of the three possibilities and its variations, providing examples. Lastly, it may be noted that Carnapian-Goguenism enjoys a computational implementation named the Heterogeneous Toolset (Hets). While it is not in scope for this essay to discuss in length about HETS, chapter 5 presents a brief overview of the ideas behind HETS and how it is structured.

The chapter that follows, named *da Costian-Tarskianism*, proposes a novel heterogeneous framework heavily inspired by Carnapian-Goguenism, drawing from da Costa pluralism and Tarski's consequence operators. The duality between Carnapian-Goguenism and da Costian-Tarskianism arises from the fact that Carnapian-Goguenism is based on institutions, which intuitively represent the semantics of logics, whereas da Costian-Tarskianism is based on *consequence systems*, which intuitively represent the syntax of logics.

The da Costian-Tarskianist framework also redefines what is an **o**ntology, albeit not with respect to a graph-theoretical construct. An **o**ntology is defined by extending consequence systems to allow encoding ontological information.

Definition 1.0.6 (extended consequence system)**.** An *extended consequence system* is a quadruple $\langle C, \mathsf{C}, C_o, \Gamma_o \rangle$, where C, C_o are signatures, C is a consequence map and $\Gamma_o \in L(C)$ is a set, such that:

1. $\langle C, \mathsf{C} \rangle$ is a consequence system
2. For every $C_k \in C$ and $C'_k \in C_o$, $C'_k \subset C_k$
3. For every $\phi \in \Gamma_o$, $\phi \in \mathsf{C}(\emptyset)$ — i.e. Γ_o is an axiomatic theory

Definition 1.0.7 (**O**ntology)**.** An ***o**ntology* is defined as a particular extended consequence system $\langle C, \mathsf{C}, C_o, \Gamma_o \rangle$. When the underlying $\langle C, \mathsf{C} \rangle$ consequence system is implicitly understood, we may drop it and refer to an **o**ntology by its components C_o and Γ_o (also called ontological aspect components).

Similarly to Carnapian-Goguenism, da Costian-Tarskianism also supports ontological refinement, integration and connection. However, in addition to the three possibilities, da Costian-Tarskianism also supports ontological *decomposition*. Ontological decomposition may be defined as the ability to extract one or more sub-**o**ntologies from a given **o**ntology, it is motivated by the intuitive idea of extracting pieces of domain knowledge from a suitably large domain and by the empirical observation that certain **o**ntologies, such as BFO, are usually described in terms of sub-**o**ntologies. Chapter 6 develops the four operations for relating **o**ntologies, thus demonstrating that da Costian-Tarskianism is a suitable candidate for ontological interoperability. Lastly, the chapter offers a detailed comparison between both Carnapian-Goguenism and da Costian-Tarskianism, with remarks that not only da Costian-Tarskianism boasts ontological decomposition as a theoretical benefit over Carnapian-Goguenism, but also that it is not restricted by relational semantics or external synthetic constructions to represent **o**ntologies.

The aim of chapter 7, entitled "*From Logic to Ontology*", is to provide an overview of logics that are in fact used to construct **o**ntologies. Notably, neither assuming the reduction thesis nor developing a heterogeneous framework answers the question of what logics are suitable for developing **o**ntologies. The chapter analyses the desirable traits of suitable logics for this task. It should be noted that most traits arise from computational concerns such as decidability and tractability of reasoning.

Furthermore, chapter 7 examines a particular class of suitable logics for developing **o**ntologies, namely the class of *description logics* (DLs), with a particular focus on the family of logics derived from $\mathcal{AL}$. The family of logics arises by extending $\mathcal{AL}$ with different role and concept constructors. For instance, the description logic $\mathcal{SROIQ}$ corresponds to $\mathcal{AL}$ extended with complements, complex role inclusion, nominals, inverse roles and qualified number restrictions, and it is the basis for the widely used representation language OWL.

The chapter also examines some of the challenges that are inherent to the field of ontological knowledge management. These challenges are particularly evident when developing systems based on ontologies, the aforementioned OBDA systems. Such challenges, as inconsistency-tolerant semantics and infinite model reasoning assumptions, have also motivated the development of non-classical description logics, such as paraconsistent and modal description logics. The chapter provides an overview on the existing literature on the various approaches tailored to address the challenges of ontological knowledge management. It presents a novel paraconsistent DL, $\mathcal{PALC}$, currently being developed in forthcoming papers by W. Carnielli, M. E. Coniglio, and Bueno-Soler [73]. $\mathcal{PALC}$, herein renamed to $\mathcal{ALC}_{\mathcal{LR}}$ for terminological reasons, serves as an example of non-classical DL aiming to solve a few of the knowledge representation and management challenges.

Chapter 7 illustrates how **o**ntologies that are represented by description logics, or languages based on description logics, may be transformed into extended consequence systems via a simple procedure. Despite not formalizing an algorithm for such transformation, the examples and natural language procedure described in the chapter serve both as proof of concept and first steps towards the development of automated tools implementing da Costian-Tarskianism.

Finally, chapter 8 presents a summary of the preceding chapters. Additionally, it discusses open questions in the fields of applied Ontology and Logic, based on the findings presented and proposals made, and suggests potential research directions. Some of the potential directions include: further developing da Costian-Tarskianism by implementing it and strengthening its theoretical foundations; automating **o**ntology construction through data processing, an area called ontology learning; integrating neural artificial intelligence with ontological reasoning, an area that is called neurosymbolic artificial intelligence; among others. The chapter, thus, concludes this essay.

Chapter 2

Ontology and ontologies

ONTOLOGY may be regarded as one of the oldest disciplines of Philosophy, next to its close relatives Metaphysics and Logic. As a first step, one could simply take Ontology to be the study of what there is. Unsurprisingly, such definition is — as it is indeed phrased — a simplification. There are a myriad of different definitions, interpretations, approaches, branches and, of course, problems of Ontology. Perhaps the most well-known ontological problem is that of whether there is a god or not, a problem for which many have provided so-called *ontological arguments* aiming to solve it. Such arguments rely on underlying ontological assumptions, which in turn may give rise to particular **o**ntologies with a lower-case O.

The objective of this chapter is to present some historical background, in hopes to clarify some current terminology. Special care is taken to focus on a particular branch of Ontology named Applied Ontology, a most pragmatic and formalized approach to Ontology concerned with specific domains of discourse.

Before proceeding, it is necessary to present an additional comment on the contents of this chapter itself. This chapter does not constitute a discourse on Ontology per se, but rather an essay on what has been called Meta-Ontology by Inwagen [190]. Two of the main meta-ontological issues are *what* Ontology itself should be about and *how* it should function. This chapter attempts to address the former through a descriptive, enumerating answer, whilst the rest of the thesis shall provide arguments for a particular answer to the latter.

2.1 Historical Survey

THE objective of this section is to provide a non-exhaustive and descriptive account of the history of Ontology. The purpose of such exposition is not to thoroughly examine or argue for or against any particular philosophical view; rather, it is to acquaint the reader with the depth of the subject and to the historical diversity of ideas. Furthermore, the reader is encouraged to consult the references cited throughout this section for more authoritative perspectives.

Historically, it may be argued that the practice of ontological inquiry in western philosophy dates back to Parmenides of Elea's sole work, his poem *On Nature*. In his poem, Parmenides explicitly regards the concepts of Being and Reality as philosophical topics for the first time in western Philosophy [201, p.1]. As with most Pre-Socratic writings, *On Nature* exists only as a reconstruction of quotations residing in subsequent works of other authors. Nineteen such quotations (labeled fragments) are currently known to exist and Parmenides postulates the existence of *being* in fragment 2 [96, p. 14]. Here is Tarán's translation of fragment 2:

> Come then, I shall tell you, and do you pay attention to the account when you have heard it, which are the only ways of inquiry which can be conceived; the one [says]: "exists" and "it is not possible to not exists", it is the way of persuasion (for persuasion follows upon truth); the other [says]: "exists-not" and "not to exist is necessary", this I point out to you is a path wholly unknowable. For you could not know that which does not exist (because it is impossible) nor could you express it. [379, p. 32]

Much is discussed about Parmenides's usage of the term εστίν in the original Greek fragment, roughly translated as "it is". In particular, questions arise as to what "it" is and what "is" refers to (e.g. existence or being or both) [96, p. 60]. Hence why Tarán [379, p. 36] simply translated it as "exists", although the traditional reading has been to understand "it" as the entirety of reality thus characterizing Parmenides as a monist. Interestingly, he explicitly identifies being as *thought*, as evidenced by the third fragment [379, p. 41].

Cordero [96, p. 64] additionally notes Parmenides regards the necessity of being as equivalent to the impossibility of non-being — "exists" and "it is not possible to not exists" —, constituting what is called *Parmenides's thesis*. The keen reader with a logical background may note some things: firstly, it seems Parmenides is informally asserting a statement and its double negation are equivalent; secondly, there is some sense of modern modality in Parmenides's thesis. These points have in fact been investigated from a logical perspective, for instance, by Marcacci [241].

Parmenides's *On Nature* gave rise to what is now called Eleaticism, a school of Ontology in which there is a single, homogeneous account of being. As per Cordero's translation of the eighth fragment:

> There is not anything to a greater degree, which would prevent its cohesion, neither is there anything to a lesser degree: it is wholly filled with that which is being. It is wholly continuous: that which is being touches that which is being. [96, p. 193]

Such account is defined by all which does not stand in contradiction to being; or, to use double negation, all that which does not not exist. To this point, Cordero [96, p. 64] observes this should prompt an analysis of the concept of non-being and an inquiry whether some things can not exist at all. Parmenides himself did not analyze what it means to be non-being as for him, "[...] it is the same to think and to be" [96, p.192, F3] and therefore there is no non-being. As a side-note, much more recently Routley [329] and Priest [311] have proposed a theory called Noneism in which things are able to, in fact, not exist within a formalized logical and metaphysical framework. This theory, in turn, is in line with Meinong's theory of objects (*Gegenstandstheorie*) wherein there are objects that do not exist.

Plato may have been one of the first to not only inspect Eleaticism in writing but also provide arguments against it. He did it directly — or rather, indirectly, as he was never featured as a character in any of his dialogues — in his aptly named *Parmenides* dialogue and extensively throughout his other dialogues, particularly in *Sophist* and *Republic*. Before proceeding, it is important to note that the existing scholarly literature on Plato is vast and ever-expanding, thus making it particularly challenging to provide a comprehensive and condensed account of his writings [308]. Therefore, this section will limit itself to an account of Plato's work in terms of what is directly pertinent to the emergence of new ontological conceptions.

In *Parmenides*, Plato presents eight hypotheses about the characteristics or nature of what he calls Unity, the monistic account of reality provided by Parmenides. He shows how the dialectical consequences of the hypotheses lead to devastating consequences. Plato's consequences are that firstly, Unity does not exist; secondly, regardless Unity of whether Unity exists, it is indistinguishable from its Eleatic counterpart (i.e. being is indistinguishable from non-being) [335, p. 48]. Furthermore, Plato refutes Parmenides's thesis by arguing that non-being is not the opposite of being; rather they are two distinct concepts that are unrelated through negation. Consequently, it is not possible to define being as the negation of non-being — for negation is not the same as opposition [103, p. 511]. This distinction between negation and opposition may be considered one of the first attempts to characterize negation, a topic which spawned a plethora of works in philosophy, language, logic and psychology and is still very much actively being discussed. The interested reader may refer to [180, 181] for a transdisciplinary account of negation and to [394] for an investigation on the geometric aspects of negation.

Plato's *de facto* ontological position actually predates both *Parmenides* and *Sophist*, yet it permeates both dialogues as a criticism of his own theory — the well-known Theory of Forms or Theory of Ideas. The Platonist Theory of Forms may be summarized as a dualistic account of reality in which there are two kinds of entities: perfect, timeless and spatially-independent entities called Forms and imperfect, time-bound and spatially-dependent entities which are instances of Forms, that is, entities which share a Form but are not Forms themselves. Crombie [103, p.253-254] investigates the chronology of Plato's dialogues and to which extent the criticisms in *Parmenides* and *Sophist* could signify Plato changed his mind about the Theory of Forms later in his life. In any case, Plato's dualistic ontology is in clear contrast to the monist Eleatic tradition.

Aristotle, Plato's pupil, not only investigated Plato's motivations into developing a non-monist ontology but also came up with an ontological position of his own. As noted by [335, p. 49], Aristotle noted one of Plato's motivations has to do with presenting an ontology which accounts for the possibility of knowledge. In fact, Aristotle was one of the first to acknowledge a key problem in Ontology, the topic of section 4 — namely, what structure should an ontological account have? Aristotle does not use this precise terminology, but rather asks of *substances* while directly addressing Platonism, in his work *Metaphysics* Λ (XII). As per Aquinas's translation of *Metaphysics* 1078b13-17:

> We must not neglect the question whether it is necessary to posit one such substance or more than one, and if the latter, how many; and we must also recall the lack of statements on this point by other philosophers, because they have said nothing about the number of these substances which can be clearly stated. The theory of Ideas makes no proper study of this problem; for the

> proponents of the Ideas say that the Ideas are numbers, and they speak of numbers sometimes as unlimited and sometimes as limited to the number ten. But as to the reason why there should be so many numbers, nothing is said apodictically. Aquinas [10, 1078b13-17]

Of course, a question arises around what exactly is a *substance*. This question has sparked much discussion and permeates nearly all ontological essays to some extent. Aristotle initially used the term substance (in Greek, οὐσία) in his earlier *Categories* to refer to the most fundamental kind in a system of ten highest kinds of being — hence why Aristotle cites the number ten in the aforementioned quote. Aristotle's ten highest kinds, dubbed *categories*, are: (1) substance; (2) quantity; (3) quality; (4) relatives; (5) somewhere; (6) sometime; (7) being in a position; (8) having; (9) acting; and (10) being acted upon [377]. A key aspect of Aristotle's categories is that each category spawns a hierarchy of lesser kinds. For instance, substance is divided into primary and secondary substances, wherein secondary substances are further divided into lesser kinds. In *Metaphysics* however, Aristotle actually shifted to a different, dyadic account of substance in terms of form and matter, while still maintaining the existence of categories [327]. Schaffer [336, p. 351] broadly summarizes Aristotelian substance as the basic, ultimate, fundamental unit of being.

There is also dispute around what exactly constitutes a *category* itself for Aristotle. As pointed out by Frede [123, p. 29] categories may be the set of beings themselves in the category, or classes of expressions of a certain kind, or yet merely concepts in and of themselves. In any case, categories may be seen as a conceptualization of general properties shared by beings, i.e. universal structures or patterns. It is in this context that Aristotelian or syllogistic logic arose, in order to formalize[1] these universal structures via definitions and axioms, allowing for the creation of ontological knowledge via constructive means (as per *Metaphysics* 9). This class-based approach to reasoning is seen by some such as Rayside and G. T. Campbell [323] as the origins of what is now called object-oriented programming in computer science.

It is evident that Aristotle's ontological account diverges from the Platonist and Eleatic views. It not only espouses a truly pluralistic stance but also exhibits a hierarchical structure. Parmenides, Plato and Aristotle present prototypical examples of the three types of ontological accounts as described by Schaffer [336, p. 354] — flat, sorted and ordered (or hierarchical), respectively.

Several contemporaries of Aristotle held opposing and agreeing views on various degrees. For instance, the Stoics (such as Zeno of Citium) agreed that substance does exist, but argued that substance is necessarily material, a vehicle for a divine active principle called *pneuma* permeating all existence [253]. Stoic philosophy posits a system of four categories: matter r substance, quality, disposition, and relative disposition or relation [326, p. 394] [253]. Pyrrho of Elis, on the other hand, rejected the concept of substances altogether [32, p. 22-23]. Despite this divergence of opinion, subsequent

[1]Here it is necessary to observe that the expression "formalize" is being used anachronistically in two senses. Firstly, Aristotle did not acknowledge formalizing his ideas, for he merely sought to present a clear way to reason about universal structures [222, p. 19]. Secondly, the current meaning of formalization is stronger than what Aristotle presented, having to do with using formal systems and formal semantics, and would not show up in the context of Ontology until much later. However, there is work to formalize Aristetolian ontology according to the usual, mathematical meaning — see, for instance, Spies and Roche [368]'s work.

thinkers of Aristotle largely concurred with his class-based view of Ontology, which led to further subdivisions up until the Middle Ages [322, p. 346] and the emergence of substance theory. In fact, Smith and Welty [361] go so far as to boldly claim that Ontology did not develop much in the 2000 years following Aristotle. However, this does not imply Ontology was left in a state of stagnation. René Descartes [101, p. 1-62], Baruch Spinoza [391], John Locke [205], Gottfried Leibniz [221], Francisco Suárez [102], Christian Wolff, among many others, had views on ontology, albeit all based on the concept of substance. Indeed, the first documented instance of the term “ontology” (in Latin, *Ontologiae*) appeared in Jacob Lorhadus’s book *Ogdoas Scholastica*, in 1606, though it did not become widely known until Christian Wolff’s *Philosophia prima sive Ontologia* in 1730 [165, p. 52].

The scientific revolution is what initiated a profound transformation from substance-based ontologies to a much broader spectrum of ontological schools, as pointed out by Smith and Welty [361]. Indeed, this significant transformation was initiated by Immanuel Kant’s *Critique of Pure Reason* [203]. Kant not only rejected substance-based ontologies, but also rejected the possibility of a single, general ontology altogether — thus acknowledging a second key problem in Ontology further discussed in chapter 4.

> The Transcendental Analytic accordingly has this important result: That the understanding can never accomplish a priori anything more than to anticipate the form of a possible experience in general, and, since that which is not appearance cannot be an object of experience, it can never overstep the limits of sensibility, within which alone objects are given to us. Its principles are merely principles of the exposition of appearances, and the proud name of an ontology, which presumes to offer synthetic a priori cognition of things in general in a systematic doctrine (e.g., the principle of causality), must give way to the modest one of a mere analytic of the pure understanding. [203, p. 358-359]

The reason why Kant’s position is so radical is that, prior to his *Critique*, Metaphysics and Ontology were used interchangeably in practice. This is evident from the definition ”Ontology” provided by Lorhadus. This led to concepts now called *external* and *internal* metaphysics. The former is concerned with the study of the world, the reality itself, while the latter comprises the study of specific theories or systems of belief. Kant and his successors argue Ontology as a science should be concerned with the internal rather than external metaphysics.

Kant’s work gave rise to a number of different currents of Ontology, both following his own ontological position and not. A few such currents will be mentioned, however none shall be examined to a great depth as such enterprise is not in the scope of this chapter. Nevertheless, section 2.2 will concentrate on two such currents of Ontology with the objective of elucidating the effective origins of Applied Ontology, which differs significantly from other branches.

Arguably, Bernard Bolzano may be the first thinker to contribute to the new ontological turn initiated by Kant [87, p. 23], presenting what Berg [35, p. 31] describes as an ontological account containing mereology, substances and abstract objects. Bolzano’s work exerted great influence on two currents of Ontology which are now collectively referred to as *Phenomenological* and *Analytical*, spearheaded by Edmund Husserl and Gottlob Frege, respectively. Franz Brentano’s contributions to Ontology have also been credited to give rise to two further branches of Ontology now called Continental and Austro-Polish, whose forerunners are Martin Heidegger [174, p. X] and Kazimierz Twardowski [386].

As previously stated, the objective of this section is not to provide a comprehensive account of the history and current trends of Ontology, as this would be a great undertaking itself. The aim is, in fact, to provide the necessary historical background for the two aforementioned key problems, which are later recalled in chapter 4. The two key problems of Ontology, in turn, relate to two overarching and opposing philosophical schools: namely, monism and pluralism. Informally, the monist ontological stance admits only a single correct ontology whereas the pluralist stance admits several correct ontologies. This distinction will be made clearer not only in chapter 4 but also in chapter 5.

For further references on the history of and contemporary essays on Ontology, the reader may turn to [5, 79, 302, 399].

2.2 Applied Ontology in Theory

FROM a taxonomic point of view, applied Ontology has its origins in a sort of diamond. It inherits concepts and features so-to-speak from the phenomenological and analytic branches of Ontology, which in turn are derived from Kantian Ontology. This section presents how two seemingly opposing branches of Ontology merged and culminated into a distinct school of Ontology by first examining the concept of "formal" or "formalized" ontologies and how it originated. This concept is fundamental to provide a set of working definitions for **O**ntology and **o**ntology in the context of applied Ontology. Finally, this section offers an account of the philosophy of applied Ontology itself.

2.2.1 Origins

EDMUND Husserl, the founder of the phenomenological school of Ontology, is responsible for first defining the term "formal ontology" in his early works, particularly in *Logical investigations* [187, 188]. As Poli [301, p.2] retells, Husserl characterizes the concept of "formal" in contrast to "material" and not in the mathematical or logical sense. For Husserl, "material" concepts precede "formal" concepts in that they arise from the realm of perception and regional ontologies, such as 'animate organism' or 'material thing' [4, p.200-201]. A "formal ontology" in the Husserlian sense is a categorization of the pure categories of objects, which apply to all domains of objects independently of the peculiarities of them. The Husserlian — and by extension, phenomenological — view is that formal ontology and formal logic differ in what they address, for formal ontology deals with categorial objects and formal logic addresses the meaning of the objects [301, p. 12]. Although the two areas differ, Husserl endorses that they are not contained in each other in any sort of hierarchy, but actually correspond to different views of reality [301, p.4]. Figure 2.1 illustrates the divide between material and formal; logical and ontological according to Husserl, adapted from [301].

Poli [300, p. 185] highlights that formal ontology does not rely on any specific formalisms and actually sits in-between two other levels of theory construction. Poli refers to the outcome of ontological inquiry as "descriptive ontology" and defines "formalized

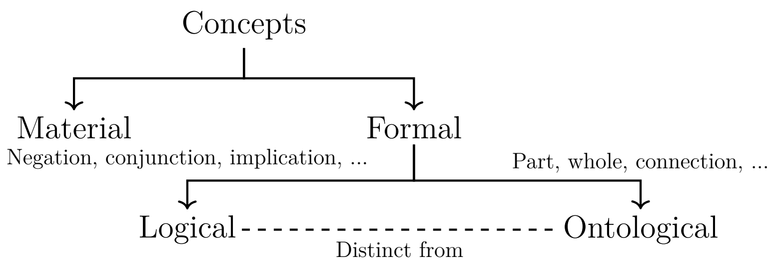


Figure 2.1: Husserlian ontological taxonomy.

ontology" as a formal ontology where there is in fact a proper formalism in the form of a codification. In order to clarify any terminological confusion, the diagram 2.2 illustrates Poli three levels [301].

$$\text{Descriptive} \xrightarrow{\text{Structure}} \text{Formal} \xrightarrow{\text{Formalism}} \text{Formalized}$$

Figure 2.2: Poli's three levels of ontology construction.

On the other hand, the analytic school of ontology has a different view on the meaning of a "formal ontology". Gottlob Frege's "Begriffsschrift" is widely regarded as the first work on analytic ontology, in which Frege provides a logic and language-oriented approach to ontology [124]. Frege held that ontology should be described by a "logically perfect language", a universal framework for thought, knowledge and reasoning — something Leibniz had previously called a *characteristica universalis* [85, p. 121]. Note the Fregean view is there exists a *single* ontology inherently formalized by logic. For analytic philosophers, Poli's three levels break down into two levels or perhaps even a single level, as an ontology in the analytical sense should have structure and be presented via formalization. To be more precise, descriptive and formal ontologies do not necessarily correspond to ontologies per se in the analytic school, as they need not be formalized by logic.

Subsequent thinkers of Frege, such as Bertrand Russell did endorse Frege's view. However, Russell's perspective diverged from Frege in that he espoused a radical radically empiricist stance, maintaining that the most basic entity in Frege's ontology (or, language) should be constituted by events corresponding to sensory experience [83, p. 6]. Russell also observed that Frege's ontology, as presented in *Begriffschrift*, is indeed subject to the widely known paradox of self-reference [85, p. 122]. An interesting point, as argued by [83, p. 7], is that neither Frege nor Russell presented an ontological account which explicitly deals with categories or classes in the sense of Plato, Aristotle, or even Kant[2]. It was in fact Rudolf Carnap, in his *The Logical Structure of the World*, who first presented a comprehensive and detailed account of how to reconstruct human knowledge by analytic means, through logical types and classes of objects defined by modalities. Carnap's ontological views have historically been subject to much discussion and confusion, as noted by Arroyo and Silva [11, p. 4]. Chapter 7 resumes a discussion on some of his philosophy.

[2]Dejnožka [111] presents an in-depth essay on Russell's ontological phases. In his essay, Dejnožka argues that Russell held a substance-oriented view of Ontology for most part and that he had great influence on Quine's stance of identity.

Despite being regarded as an analytic philosopher, Willard Van Orman Quine was one of the first to challenge the analytic view on language or logic as ontology in his seminal paper "Two Dogmas of Empiricism" [315]. Quine's work blurred the boundaries between analytic and non-analytic (and by extension, phenomenological) ontology. In addition, Quine indirectly introduced another pivotal concept in Applied Ontology: namely, that of *ontological commitments*. He did so by denying what Frege and Russell termed *universals*, abstract entities which stand for ontological properties or classes themselves (e.g. attributes, numbers, properties), in his seminal essay "On What There Is" [314]. By rejecting universals, Quine presented a method to determine what entities an ontology is committed to:

> The issue is clearer now than of old, because we now have a more explicit standard whereby to decide what ontology a given theory or form of discourse is committed to: a theory is committed to those and only those entities to which the bound variables of the theory must be capable of referring in order that the affirmations made in the theory be true [314, p.33].

In other words, Quine says an ontology is committed *only* to what it can refer to, therefore endorsing a sort of "local" context for ontologies, in plural. He is in fact, very much vocal about acknowledging the plurality of ontologies and even remarks a point of chapter 4; if there are multiple ontologies, how to compare them? This is Quine's inquiry:

> Now how are we to adjudicate among rival ontologies ? Certainly the answer is not provided by the semantical formula "To be is to be the value of a variable"; this formula serves rather, conversely, in testing the conformity of a given remark or doctrine to a prior ontological standard. We look to bound variables in connection with ontology not in order to know what there is, but in order to know what a given remark or doctrine, ours or someone else's, says there is; and this much is quite properly a problem involving language. But what there is is another question [314, p. 34-35].

It may be noted that Quine's ontological commitments correspond to the third of three tenets underlying the Quinean meta-ontological framework. As summarized by Varzi [388], the Quinean tenets or credos are:

1. There is only one notion of existence, adequately captured by the existential quantifier;
2. Being and existence are the same;
3. We are ontologically committed to all and only those entities that must exist in order for the theories or statements we hold to be true to be true.

For Quine, the "existential quantifier" which adequately captures existence is not any first-order quantifier: it is, necessarily, the existential quantifier from first-order classical logic. Smid [356] discusses the reasons behind Quine's choice to elect first-order classical logic as the one and only language which correctly characterizes existence. Quine's reasons may be briefly summarized as two overarching arguments: firstly, that first-order classical logic is as powerful as any logic can be while enjoying a complete proof procedure for validity and inconsistency; secondly, that first-order classical logic is as ontologically innocent as a logic could possibly be, precisely because first-order classical logic is complete and, thus, logical truth may be defined proof-theoretically. However, as the subsequent sections 2.2.2 and 2.2.3 illustrate, Quine's conservative views have become increasingly unfashionable in the context of contemporary, applied Ontology. Furthermore, chapter 4 expands on this point from the philosophical point of view, whereas chapter 5 examines practical reasons why it is the case.

Inspired by Poli's aforementioned taxonomy and based on Quine's historical classification in [314, p. 33-35], figure 2.3 attempts to present Frege-Russell's radically different analytical taxonomy of ontology to contrast Husserl's phenomenological taxonomy. The diagram illustrates that all concepts are derived from the formal domain, however the ontological domain spawns concepts stemming from perception. Indeed, despite the fact that "Concepts" is not the root of the taxonomy, the Frege-Russell school does not endorse a fully formalistic school of ontology for logic (and formal language) is seen as the means to *discover* or arrive at abstract concepts.

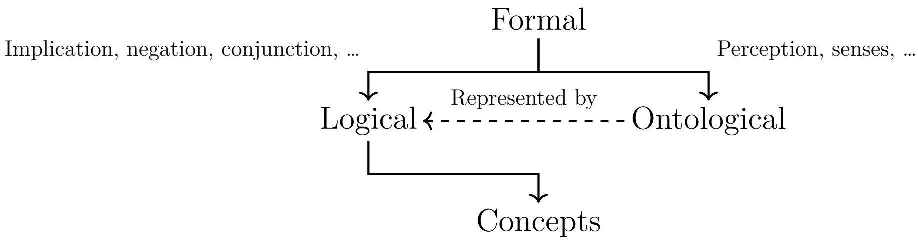


Figure 2.3: Frege-Russell's ontological taxonomy.

How might then one combine or at least accommodate for the seeming dichotomy in the analytical versus phenomenological debate? The answer lies in pioneering work of both Nino Cocchiarella [84] and Barry Smith [357]. Cocchiarella developed, through the course of several years, what he termed Conceptual Realism, a formal ontology based on linguistic and logical analysis of acts via reference and predication, not relying on any phenomenological reductions[3]. Although he is regarded as an analytic philosopher and in fact refers to his own theory as such, Cocchiarella acknowledges Conceptual Realism is indeed based on what he calls "our scientific knowledge of the world" [83, p. 21] and must account for certain phenomena (such as time and thought) [83, p. 25-26]. Therefore, one may argue Conceptual Realism, analytic as it may be, has certain phenomenological traits built into itself — namely that of relying on a descriptive ontology and using logic to describe key phenomena.

Smith, on the other hand, is not so quick to categorize himself as analytic. He takes the Quinean approach and does not categorize himself as either analytic, synthetic or phenomenological, with Poli [301, p. 188] stating he lies midway between the analytic and phenomenological traditions. Smith and Mulligan [359] presents a formal ontology based on mereology and a non-logical language, and an *application* of such ontology. Although this application is rather abstract, it sparked what eventually became a series of papers

[3]The term "phenomenological reduction" refers to a method developed by Husserl to describe the process of understanding an experience or phenomenon. Küng [210] writes on the meaning and understanding of Husserlian phenomenological reduction. The interested reader may also refer to [267] for an account of research methods derived from phenomenological reduction.

and contributions to Applied Ontology proper. In particular, Grenon and Smith [147] developed the Basic Formal Ontology (BFO), a top level ontology which will be discussed in chapter 3, with longstanding interdisciplinary impact.

To fully comprehend how Applied Ontology in fact came to be, it is essential to undertake a brief excursion into the area of Artificial Intelligence (AI), focusing on a particular a subfield of it called Knowledge Representation (KR). Contemporaneously to the pioneering works of Cocchiarella and Smith, many AI researchers began taking an interest in knowledge representation for machine reasoning. This led to the development of formalization methods focused on machine-oriented representation, the earliest work being that of Newell, Shaw, and Simon's language-based approach and J. A. Robinson's logic-based approach [276, 328].

Notably, McCarthy and P. Hayes [251] recast several problems of knowledge representation under the light of epistemology and metaphysics, such as what it means for a representation to metaphysically adequate, and proposed a logic-based formalism for what they called an epistemologically adequate system. In their paper, McCarthy and P. Hayes defined what is referred to as the Frame Problem — how may we encode, in common-sense and logic-based knowledge representation, the fact that the world remains unchanged until certain events trigger change? Attempting to answer the Frame Problem drove further kinds of representation methods, such as *frames* as proposed by Minsky [257] and Brachman [51]. The methods of formalization derived from knowledge representation and the acknowledgment of the Frame Problem mark the beginnings of ontologies in the sense of Applied Ontology, paving way to those described in chapter 3. As a matter of fact, one of the earliest appearances of the term "ontology" as a logic-based formalization of a specific domain (i.e. common-sense knowledge), committed only to that specific domain, is P. J. Hayes's ontology for liquids as in physics [109, 172]. On the other hand, Cyc is one of the first (and perhaps only) large-scale instances of an ontology in the same sense, whose construction has in fact not concluded yet [225].

Applied Ontology proper, viz. as it is currently regarded, finally culminated in the works of Gruber and Guarino [149, 150, 156], something Guarino [153] himself acknowledges on behalf of the research community. Even though the term "ontology" had been seeing usage in the context of Computer Science prior to Gruber [150], not only in AI [361, p. v] but also in database systems and information modeling [165, p. 57], past research did not give any particular importance to how the term was used. As per Guizzardi [165, p. 58]:

> [...] none of these efforts took ontology seriously, in the sense that the choices of categories that are part of the conceptualization underlying these languages were not based on Ontology in the philosophical sense. Guizzardi [165, p. 58]

As a matter of fact, Smith and Welty [361] and Guizzardi [165, p.58] both point that haphazardly creating ontologies with no philosophical grounding as being the underlying source of many problems permeating information systems, conceptual modelling and computer science as a whole to this day. On this topic, Stonebraker writes an essay on data integration and its issues [374].

Gruber [150] explicitly redefined the term "ontology" in a computational context and first expressed concerns which fundamentally shaped applied ontology — namely, *portability* and *complexity* of ontology representations. His definition and concerns are the starting point for the next section.

2.2.2 Working definitions

BY summarizing the past section, one could firstly define applied Ontology as the sub-branch of Ontology married to artificial intelligence, computer science and logic. While applied ontology is closely linked to these areas of knowledge, it is merely linked and not fully intertwined. According to Guizzardi [165, p. 56], applied Ontology concerns itself with "formal ontological theories that can be developed and applied in the solution to problems in the fields of computer and information sciences and, in particular, of conceptual modeling".

Gruber [150] defines the starting point of applied Ontology's subject matter in his first definition of what is indeed an ontology:

Definition 2.2.1 (Ontology by Gruber)**.** An *ontology* is an explicit specification of a conceptualization.

Notice the definition is rather high-level and makes no mention of logic or frame-based representations, nor makes any artificial intelligence or machine reasoning remarks. Later on, Borst [46] and Studer, Benjamins, and Fensel [376] expanded this definition into two slightly more encompassing definitions:

Definition 2.2.2 (Ontology by Borst)**.** An *ontology* is a formal specification of a shared conceptualization

Definition 2.2.3 (Ontology by Studer et. al)**.** An *ontology* is a formal, explicit specification of a shared conceptualization.

All of these definitions share the same vagueness about terms *conceptualization*, *explicit specification* and *shared*. This is a point tackled by Guarino, Oberle, and Staab [160] and Neuhaus [273], who decidedly scrutinize each term in order to yet again re-define *ontology* in a much more precise sense through means of a set-theoretical description parameterized by a logical language. In essence, Guarino, Oberle, and Staab [160] present a re-contextualized and updated interpretation of themes and definitions espoused in several of Guarino's earlier works [155, 156, 162].

Guizzardi [165] notes the motivation for Guarino's work on providing a formal definition of ontology indeed stems from the vagueness in Gruber's, Borst's and Studer, Benjamins, and Fensel's definitions, which allows for a broad spectrum of specifications to be classified as "ontologies".

In order to present Guarino's formal definition of "ontology", one must first define a few fundamental blocks, very much in a manner that is done by Guarino, Oberle, and Staab [160] and Guizzardi [165]. Namely, the necessary blocks are conceptualizations and specifications, and in particular, the distinction between extensional and intensional conceptualizations.

Definition 2.2.4 (Extensional relational structure)**.** An extensional relational structure (a *conceptualization* according to M. R. Genesereth and Nilsson [133]) is a tuple $\langle D, \mathbf{R} \rangle$ where:

- D is a set called the universe of discourse
- $\mathbf{R}$ is a set of relations over D

As Guarino, Oberle, and Staab [160] note, this requires one to *explicitly* identify the existing entities and define their relations based on a particular state of the world. This has the following implications:

1. If any change happens to D leading to a different universe of discourse D', one may potentially have to redefine every $R_i \in \mathbf{R}$, giving rise to yet another conceptualization $\langle D', \mathbf{R}' \rangle$
2. Therefore, this definition relies on particular observations of world states instead of underlying concepts permeating the universe of discourse [160, p. 5].

In light of these restrictions, Guarino [155] proposed a different definition of *conceptualization* based on the observation that when reasoning about concepts, it is much more convenient to talk about the *intended* characteristics of a certain concept instead of explicitly identifying all that it consists. For this reason, he called this new definition an *intensional relational structure*, a minimal framework to reason about the *intended* characteristics of conceptualizations whilst considering all possible world-states, despite not forcing one to explicitly define such states [165, p. 80].

In order to clearly state Guarino's intensional definition, it is necessary to define worlds and world-states. As the reader acquainted with non-classical logics will note, and indeed as it is remarked by Guarino, Oberle, and Staab [160], the terminology is quite similar to Kripke-style possible-world semantics by no coincidence.

Definition 2.2.5 (World and world-states)**.** Given a specific system[4] S being modeled, a **world-state** w is a maximal observable state of affairs, a unique assignment to all variables which characterize the system. A **world** $\mathfrak{W} = \langle W, \prec \rangle$ then is a totally ordered set of world-states W, where the order $\prec$ corresponds to S's evolution in time. If time is abstracted away, a world-state coincides with a world.

World-states and worlds are used to define what is an *intensional relation*, also called *conceptual relation* by Guarino.

Definition 2.2.6 (Intensional relation or *conceptual relation*)**.** Let S be a system, D an arbitrary set of distinguished elements of S and W the set of world-states of S. We shall call the tuple $\langle D, W \rangle$ the *domain space* of S. An intensional relation (or conceptual relation) ρ^n over $\langle D, W \rangle$ is a total function $\rho^n : W \to 2^{D^n}$ mapping a world-state to an n-ary relation on D.

A possible intuitive understanding behind the definition above is an intensional relation ρ^n allows one to "query" how a world-state maps to a domain configuration, as a world-state may include more information that what is assumed in the domain. What follows is Guarino's definition of *conceptualization*, in the intensional framework..

[4]Guarino, Oberle, and Staab [160] deliberately use the terms *system* and *variable* in a loose sense. To increase preciseness, one could define a system S as a set of sets $\mathcal{T}_i$ of totally orderable tuples T_j, that is $S = \{\mathcal{T}_1, \mathcal{T}_2, \ldots\}$. Each element of a tuple T_j is then a variable of the system, which does not necessarily correspond to an element of domain D.

Definition 2.2.7 (Intensional relational structure or *conceptualization*)**.** An *intensional relational strucutre* (or *conceptualization* according to Guarino) is a triple $\mathbf{C} = \langle D, W, \mathfrak{R} \rangle$ where:

- D is a set called the universe of discourse
- W is a set of world-states (also called *possible worlds*)
- $\mathfrak{R}$ is a set of conceptual relations on the domain space $\langle D, W \rangle$

Definition 2.2.7 concludes Guarino's take on what is a *conceptualization.* As per Studer, Benjamins, and Fensel [376], a conceptualization alone does not comprise an ontology, for it still lacks a way to specify it. For instance, one could describe the set of conceptual relations $\mathfrak{R}$ of a conceptualization using natural language, say in English, a domain-specific language, or a logical language. In any case, one may build a specification of a conceptualization by using a language **L**. Guarino [155, p. 8] defines "formal" as being "machine readable" and thus rules out natural languages from being used for formal specifications.

A language **L** is said to be *committed* to a conceptualization if it specifies such conceptualization. Guarino [155, p. 9-10] provide definitions for ontological commitments in the extensional and intensional sense, although only in the intensional sense is a specification called an *ontological commitment* per se.

Fix **L** to be some variant of a first-order logical language, with purely relational vocabulary **V** — that is, a set of constants and predicate symbols, but no function symbols.

Definition 2.2.8 (Extensional first-order structure)**.** Let **L** be a first-order logical language with relational vocabulary **V** and $S = \langle D, \mathbf{R} \rangle$ an extensional relational structure. An *extensional first-order structure* (or *model* for **L**) with respect to a system S is a tuple $M = \langle S, I \rangle$ where I (called an *extensional interpretation function*) is a total function $I : \mathbf{V} \to D \cup \mathbf{R}$ mapping each $v \in \mathbf{V}$ to either an element of domain D or an extensional relation $R_i \in \mathbf{R}$.

The reader familiar with model theory will note this is a very standard definition of structure for first-order languages, as is presented for instance in Hedman [173, p. 59]. Guarino extends this definition to the intensional case.

Definition 2.2.9 (Intensional first-order structure or *ontological commitment*)**.** Let **L** be a first-order logical language with relational vocabulary **V** and $\mathbf{C} = \langle D, W, \mathfrak{R} \rangle$ an intensional relational structure (or, equivalently, a conceptualization). An *intensional first-order structure* for **L** is a tuple $\mathbf{K} = \langle \mathbf{C}, \mathcal{I} \rangle$ where $\mathcal{I}$ (called *intensional interpretation function*) is a total function $\mathcal{I} : \mathbf{V} \to D \cup \mathfrak{R}$ mapping each $v \in \mathbf{V}$ to either an element of domain D or an intensional relation $R_i \in \mathfrak{R}$.

Intuitively, the term ontological commitment is used as a reference to Quine's ontological commitments in how they are used to restrict what ontologies are able to refer to. Indeed, ontological commitments may be used to constrain models of a given language, giving rise to the concept of *intended models.*

Definition 2.2.10 (Intended models)**.** Let $\mathbf{C} = \langle C, W, \mathfrak{R} \rangle$ be a conceptualization (as per 2.2.7), $\mathbf{L}$ a first-order logical language with vocabulary $\mathbf{V}$ and $\mathbf{K} = \langle \mathbf{C}, \mathcal{I} \rangle$ an ontological commitment. A model $M = \langle S, I \rangle$, with $S = \langle D, \mathbf{R} \rangle$ is an *intended model* according to $\mathbf{K}$ iff:

1. For all constant symbols $c \in \mathbf{V}$, $I(c) = \mathcal{I}(c)$
2. There exists a world-state $w \in W$ such that, for each predicate symbol $v \in \mathbf{V}$, there exists an intensional relation $\rho \in \mathfrak{R}$ such that $\mathcal{I}(v) = \rho$ and $I(v) = \rho(w)$.

The set $\mathbf{I_K}(\mathbf{L})$ of all models of $\mathbf{L}$ that are compatible with $\mathbf{K}$ is called the set of *intended models* of $\mathbf{L}$ according to $\mathbf{K}$.

Condition 1 of the above definition states constant symbols need to be mapped to identical elements of the universe of discourse. Condition 2 states there must be a world-state w in which each every predicate symbol v is mapped to an intensional relation ρ whose value $\rho(w)$, at that world-state, coincides with the extensional interpretation of the predicate $I(v)$. In other words, there must be a world where intensional and extensional counterparts coincide.

The purpose of intended models is not to match a given conceptualization, but rather to bound the interpretation of such conceptualization to a certain perimeter of models. As per Guarino [155]:

> A set of intended models is therefore only a weak characterization of a conceptualization: it just excludes some absurd interpretations, without really describing the "meaning" of the vocabulary [155].

In [155, p. 5], Guarino provided yet another revised, natural language definition of **o**ntology stemming from intended models. What follows is the slightly shortened and modified version of the same natural language definition encountered in [160, p. 11], for this version emphasizes more clearly Guarino's view of what constitutes an **o**ntology is functional, viz. it is based on what is an **o**ntology's purpose.

Definition 2.2.11 (Ontology by Guarino et. al)**.** An *ontology* is a logical theory designed to account for the intended meaning of the vocabulary used by a logical language.

5 later revisits the matter of defining ontologies in a much more pragmatic sense, but for now the term **o**ntology (with a lower-case O) in the context of applied Ontology may be understood as presented in the formal definition below.

Definition 2.2.12 (Ontology)**.** Let $\mathbf{C}$ be a conceptualization, $\mathbf{L}$ be a logical language with vocabulary $\mathbf{V}$ and $\mathbf{K}$ be an ontological commitment. An ontology $\mathbf{O_K}$ for $\mathbf{C}$ with vocabulary $\mathbf{V}$ and ontological commitment $\mathbf{K}$ is a logical theory consisting of a set of formulas $\mathbf{L}$ designed so that the set of its models approximate as well as possible $\mathbf{I_K}(\mathbf{L})$, the set of intended models of $\mathbf{L}$ according to $\mathbf{K}$.

Guarino's definition of **o**ntology is, intuitively, a logical theory which attempts to capture ontological commitments. It reverses the roles of ontological commitments when compared to Gruber [150]'s view — ontologies arise from ontological commitments and not vice-versa. Intended models commit to a certain conceptualization and an ontology in the current sense reflects this commitment by attempting to approximate its intended models. In other words, ontology specification strives to internalize ontological commitments as much as possible, however there may be a gap between the set of intended models and the set of models which satisfy the logical theory of an ontology. This may be visualized in figure 2.4 below, where the set of models which satisfy the ontology $\mathbf{O_K}$ does not match the set of intended models $\mathbf{I_K}(\mathbf{L})$.

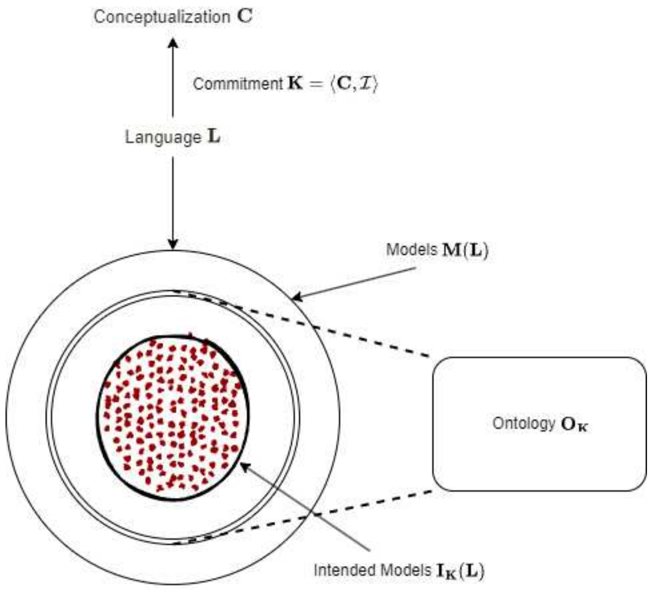


Figure 2.4: How an ontology arises as a specification of a conceptualization. From [155, p. 5] and [164, p. 8].

Notice Guarino's definition is not prescriptive, viz. it does not impose any sort of restrictions nor does it provide a process through which one may decide which domain, vocabulary or language to use. Furthermore, the definition of ontologies places them in a sort of conceptual vacuum. It does not provide a *direct* framework or way for ontologies to interact with each other, aside from perhaps the underlying ontological interrelations that may be drawn at the meta-theoretical level through set theory. For instance, one could say a proper sub-ontology $\mathbf{O'_K} = \mathbf{L'}$ of an ontology $\mathbf{O_K} = \mathbf{L}$ according to Guarino is such that $\mathbf{L'} \subset \mathbf{L}$.

Additionally, it is imperative to note that Guarino's definition *does not* state that $\mathbf{L}$ is, exclusively, first-order logic. This illustrates that existence in applied Ontology is not contingent on any dogmatic imperatives such as what Quine had argued for. Furthermore, it is perhaps inconsistent that Guarino's definition (and, by extension, many definitions built upon it) utilizes Quine's ontological commitments as its foundations, but rejects Quine's view on first-order logic as the de facto language for representing existence. However, it is not contradictory. This is because an ontology, as per Guarino, does not aim to represent reality in its entirety and, as such, may accommodate or represent several forms of existence.

Despite the aforementioned lack of prescriptivity and internalized tools for ontology interaction in definition 2.2.12, the third tenet of the original definition of **o**ntology (i.e. definition 2.2.3) is the *sharing* aspect of conceptualization. Thus, the matter of ontological interaction is an integral part of Applied Ontology, and it has been acknowledged as such since the foundational period of the area, including by Guarino [155, 158], Gruber [150] and Bylander and Chandrasekaran [58]. Guarino, Oberle, and Staab [160] rely on the semiotic triangle devised by Ogden and Richards [281], based on work by Saussure, Frege and Peirce, to explain why conceptualizations must be shared — the main point being shared vocabulary alone is not enough to ensure clear reasoning in a multiagent scenario. Thus, although definition 2.2.12 places ontologies in individual vacuums, the process of constructing a *particular* **o**ntology requires consensus at the meta-ontological level. In fact, this is what the expression "approximates as well as possible" is encapsulating.

Chapters 3 and 4 revisit the topics of choice of language, vocabulary and domain, alongside the matters of ontology integration, interoperability and granularity. For the time being, definition 2.2.12 suffices as a starting point to specify what is the subject matter of applied Ontology.

2.2.3 On the philosophy of applied Ontology

Before heading into the next chapter, it is necessary to examine the philosophy of applied Ontology. The reason for doing so is to clearly set applied Ontology apart from other similar-sounding branches of Ontology. This section, then, doubles as a preface to understand the mindset and philosophical assumptions behind the **o**ntologies to described in the forthcoming chapter.

Perhaps an initial question one may ask is whether applied Ontology as practiced is still tethered to any particular philosophical underpinnings. There are indeed varying degrees of explicit philosophical concern in papers in the area, ranging from none (such as more machine-oriented and domain-focused works, see for instance [14, 209, 395, 398]) to foundational works such as previously cited [46, 150, 155, 361, 376]. But is there a common philosophical aspect shared among the literature? Bihan and Barton [40] argue this question has a positive answer[5]:

> Applied ontologies are often based on metaphysical principles that are not inspired by contemporary physics. These principles are sometimes closer to common sense (and the naïve physics coming with it) or classical physics. [...]
>
> Other applied ontologies do not rely as explicitly on human common sense but aim at providing an ontology compatible with some special sciences such as biomedicine, where the first aim is not to develop metaphysical principles inspired by contemporary physics. This is due to the fact that for their largest part, many special sciences such as biomedicine do not require the use of physical models more elaborate than naïve or classical physics. [40, p. 4]

To put the quote above into context, Bihan and Barton are interested in providing an argument for the usefulness of analytic metaphysics by extending the *heuristic value* argument of French and McKenzie [126]. Hence, the term "metaphysical principles" in the quote refers to are principles from analytic metaphysics. Regardless of the dispute surrounding analytic metaphysics, one may check that indeed none of the previous definitions

[5]In this excerpt, the term "applied ontologies" refers to **o**ntologies in the sense defined in the previous section.

of **o**ntology presented in the previous section considered scientific standards. Bihan and Barton [40] further dissect another philosophical position permeating applied Ontology:

> [...] an ontology built upon a non-physicalist methodology might facilitate computations. [...] using somewhat simpler metaphysical principles from analytic metaphysics (rather than a description based on contemporary physics) might be a way to ensure that the logical consequences remain computable in a reasonable time for practical purposes. Additionally, these principles enable human users to browse and use the ontology more easily than if it was based on e.g. quantum mechanical or relativistic principles. [40, p. 8]

The literature on applied Ontology is also generally pragmatic, as the prioritization of computational tractability is a primary concern. This is highlighted by Grenon and Smith [147]:

> [...] a line should be drawn between external reality and whatever our systems of private or collective representation might be. The focus of ontology is not the latter, but the former. But there is nonetheless some truth behind the motivations of the pragmatist conceptualist: for one thing, we do want our representations to be useful; for another, people do actually differ and disagree in their representations of the world. [146, p. 78]

It is evident, then, that applied Ontology incorporates elements from both (analytic) metaphysics and conceptual pragmatism. One may also inquiry whether applied ontologists respect the traditional epistemology vs. ontology divide. Although **o**ntologies are devoid of epistemological content[6], the process through which they are built across the literature of applied Ontology is not. This assertion is not unexpected, as it is consistent with the "shared" tenet of **o**ntology construction. This is evidenced once again by Grenon and Smith [147]:

> [...] the ontologist who is concerned with reality must make provisions for the evolution and refinement of the views underlying her work. [...] We must be ready to abandon views and introduce unforeseen elements, even if this requires us to redo laborious work.
>
> [...] as ontologists, our methodology must also make provisions for the possibility of maintaining equally legitimate realist perspectives on reality. [147, p. 79-80]

In broad strokes, one may characterize the philosophical views shared by the majority of the applied Ontology as clearly being still tied to metaphysics (albeit clearly not in a traditional sense as described in section 2.1, being much closer to analytic views), and guided by both conceptual pragmatism and realist adequatism and perspectivalism [40, 147, 250]. The interested reader may refer to [147] and the third chapter of [165] for a much deeper foray into the philosophy of ontological analysis in the context of applied Ontology.

[6]This, again, relates to the fact that **o**ntologies merely attempt to capture models ascribed by ontological commitments.

In conclusion, it is imperative to issue a terminological warning. Applied Ontology is not to be confused with “Scientific Ontology”, a term coined by Chakravartty [78]. Applied Ontology is clearly oriented towards a formal, machine-oriented treatment of its subject matter (i.e. **o**ntologies), driven by the recently mentioned philosophical views. In contrast, Scientific Ontology is an account of Ontology based naturalized metaphysics, a distinct branch of Ontology. In addition, in certain contexts, the term “applied Ontology” is used to refer to a concept that is, in fact, *not* applied Ontology at all. For instance, Jacquette [196] uses the terms “applied Ontology” and “applied scientific Ontology” interchangeably — both being much closer to analytic Ontology than to applied Ontology in the current sense. It is perhaps fortuitous that such a semantic discrepancy has arisen, as it serves to illustrate the limitations of natural language as a means of developing **o**ntologies that can guarantee interoperability.

Chapter 3

From Metaphysics to Industry

THE following quote may be considered an adequate introduction to the chapter's objectives:

> We find it remarkable that an activity that traces its origins to the work of philosophers who lived more than two millennia ago has become central to the development of modern information technology. [159, p. 2]

The "activity" Guarino and Musen refer to is, of course, Ontology in the sense that is described in the historical remarks of chapter 2. This chapter will examine the ways in which applied Ontology has become a central concept in modern information technology. It will present a number of concrete instances of **o**ntologies currently being used in several domains, including industry, science, and academia. It will also inspect what choices of domain, language each **o**ntology possesses and what ontological commitments they are bound to. Such inspection will demonstrate the relationship between theoretical concepts, as presented in the previous chapter, and their applications.

3.1 A taxonomy of ontologies

GIVEN the absence of universal consensus or enforced choice of language, domain, and vocabulary, it is evident that there exist a multitude of **o**ntologies in the literature. This observation, coupled with the problem of integrating different **o**ntologies, has led Guarino [155, p. 7-8] to propose a taxonomy of **o**ntologies based on their level of generality, illustrated in figure 3.1:

- *Top-level **o**ntologies* describe the most general concepts such as space, time, matter, object, event, action, among other. These concepts are independent of a particular problem or domain. Guarino suggests using unified top-level **o**ntologies in cases where there is a large community of users.
- *Domain and task **o**ntologies* describe the vocabulary related to a generic domain (like medicine, automobiles, law) or a generic type or activity (such as diagnosing, selling, judging) by *specializing* terms introduced in a top-level **o**ntology.

- *Application* ***o****ntologies* describe concepts specific to a particular domain and task, usually consisting of specializations of both related **o**ntologies. The concepts correspond to the roles played by domain entities performing a certain activity, here are some examples:
    - *autoimmune disease* in the domain *medicine* and task *diagnosing.*
    - *spare component* in the domain *automobiles* and task *building.*
    - *lawsuit* in the domain *law* and task *judging.*

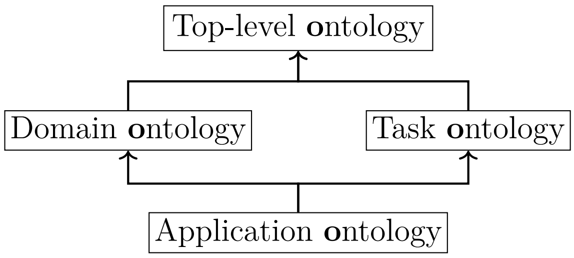


Figure 3.1: Guarino's taxonomy of ontologies, extracted from [155] (and originally presented in [157]).

The computer science oriented reader may initially regard application **o**ntologies as being merely data or knowledge bases, where each entity is a row of a database or a node in a knowledge base. This is not the case, for application **o**ntologies distinguish themselves for two reasons. Firstly, the purpose of **o**ntologies is to represent state-independent information whereas data and knowledge bases store both state-independent and state-dependent information (such as a list of currently known diseases or active lawsuits). Secondly, for the purpose of an **o**ntology is also to capture the semantics behind concepts and provide a framework for reasoning about them. Conversely, databases are concerned with the systematic storage of structured data. Furthermore, knowledge bases may be regarded as a *superset* of **o**ntologies and databases, as they allow for reasoning and storage of structured data simultaneously [155, p.8] [199].

The next few sections present examples of top-level, domain, and task **o**ntologies. The deliberate choice to omit application **o**ntologies lies in how they differ from task or domain **o**ntologies.

Application **o**ntologies are more finely grained than domain **o**ntologies, and are tailored to the specific needs of a particular application, such as healthcare, manufacturing, or finance. In general, application **o**ntologies utilize concepts and relationships from domain **o**ntologies while also including additional concepts and constraints that are specific to the application context. For example, an **o**ntology of sports designed to capture concepts and relationships between sports-related concepts, across all existing sports such as football, basketball, volleyball etc. would be considered a domain **o**ntology. An **o**ntology of football based on the **o**ntology of sports, with additional concepts and relationships specific to football, would comprise application **o**ntology. In summary, the distinction lies in the level of specialization and granularity. As the focus of this chapter is directed towards issues concerning applied Ontology as a whole, it is not within scope of this discussion to address the specific of application **o**ntologies.

There are, however, outliers which do not conform to Guarino's taxonomy. A domain **o**ntology does not necessarily need to be derived from a top-level **o**ntology — as previously remarked, the definition of **o**ntology allows for an "ontological vacuum" of sorts in which **o**ntologies can freely exist without being aware of each other. One possible term for **o**ntologies which do not conform to Guarino's taxonomy is *deviant **o**ntologies*. Deviant **o**ntologies can exist at any level of the taxonomy and a particular example will be described in section 3.4.

Another crucial concept in the field of applied Ontology is that of an *ontological foundry*, which directly relates to the "shared" aspect of ontological analysis. An ontological foundry or simply foundry is an organization which maintains and develops a collection of **o**ntologies, deviants or not, focusing on maintenance and ontological quality. Foundries address the aforementioned "ontological vacuum" by developing tools that facilitate collaboration and interoperability of **o**ntologies. The number of **o**ntologies managed by a foundry can range from a single one to more than a hundred [171, 209, 269, 362]. A few examples of ontological foundries are:

- The Open Biological and Biomedical Ontologies (OBO) Foundry [362].
- The Industrial Ontologies Foundry (IOF) [209].
- The Ontology-Driven Data Documentation for Industry Commons (OntoCommons) [2].

Figure 3.2 proposes an extension of Guarino's taxonomy, which considers deviant **o**ntologies and foundries.

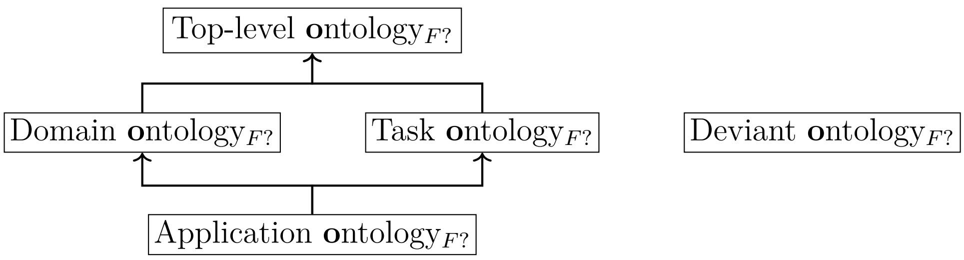


Figure 3.2: An extension of Guarino's taxonomy of ontologies. The subscript $F?$ denotes an **o**ntology optionally belongs to a foundry.

3.2 Top-level ontologies

Top-level **o**ntologies, also called upper or foundational **o**ntologies, attempt to provide an answer to the following foundational question — in a classification of reality, what is the structure of the very top level? [198]. In answering this question, top-level **o**ntologies provide a useful base or foundation of categories[1] upon which to build domain and task **o**ntologies, usually facilitating their integration in a practical application [303, p. 573]. Top-level **o**ntologies are particularly biased in a sense, for the ontological commitments

[1]A category being an overall "class of entities", a distinguished set of the domain of a given ontology.

they capture are very broad and often correspond to assumptions about the nature of categories themselves.

As Jansen [198] notes, the very first top-level ontology (in the traditional philosophical sense) is Aristotle's list of categories, for it is an attempt to characterize all reality via classes of entities. It can be reasonably argued that every ontology described in the historical subsections of chapter 2 can be considered a top-level ontologies. This is because the philosophical discipline of Ontology has always sought to address the fundamental question of what exists. Nonetheless, none of the aforementioned ontologies align with the definition of **o**ntology, as they were never presented through a formal representation language.

This section presents top-level **o**ntologies which are in fact compliant to Guarino's definition, and it seeks to familiarize the reader with the array of responses to the aforementioned foundational question. The objective of this section is not to examine the philosophical presuppositions and ontological commitments of each **o**ntology. However, it is necessary to briefly discuss such topics when presenting the classes an **o**ntology contains. For a treatise dedicated to the philosophical principles underlying in top-level **o**ntologies, the reader may refer to [198]. Gangemi et al. [130] and J. F. Sowa [367] analyze how choosing a philosophical principle concretely affects the task of building a top-level **o**ntology. Lastly, the reader may refer to the thorough survey [287] dedicated to top-level **o**ntologies.

Cyc

The Cyc top-level **o**ntology consists of not only a top-level **o**ntology but an overall project whose aim is capture common-sense knowledge and reasoning, covering all things from abstract ideas to concrete entities. The Cyc project started in the 1980s led by Douglas Lenat and has been under active development since, although now under guidance of the Cycorp company [358]. Cyc is a closed source **o**ntology, meaning one cannot examine its implementation details to understand its structure, but an open variant named OpenCyc was released by Cycorp in 2002 as a subset of its closed counterpart. As of 2021, Cyc contains around 1.5 million concepts (i.e. a vocabulary of 1.5 million terms and relations) and around 25 million rules to reason about those concepts (i.e. the ontological theory consists of at least 25 million axioms) [104, p. 6].

Cyc is not particularly opinionated as far as its actual "top-level" structure goes, with no explicit ontological commitments. One key relation in Cyc's vocabulary is the "is-a" relation, for every concept in Cyc descends from a top `Thing` concept via this relation. The "is-a" relation allows for multiple inheritance as can be seen from the snapshot of Cyc's top-level structure in figure 3.3 [287, p. 85-86].

The representation language of choice for Cyc is called CycL. CycL is not a logic per se, such as first-order logic or first degree entailment [282], but rather a language heavily influenced by the programming language Lisp and predicate calculus. Initially, CycL started out as an extension of RLL, a frame-based[2] Representation Language Language, also developed by Douglas Lenat [145], but it has since been rebuilt to increase expressive

[2] A frame is a structure initially devised by Minsky [257] to represent situations in a machine-readable format. Minsky was originally interested in the problem of representing knowledge purely in the context of artificial intelligence.

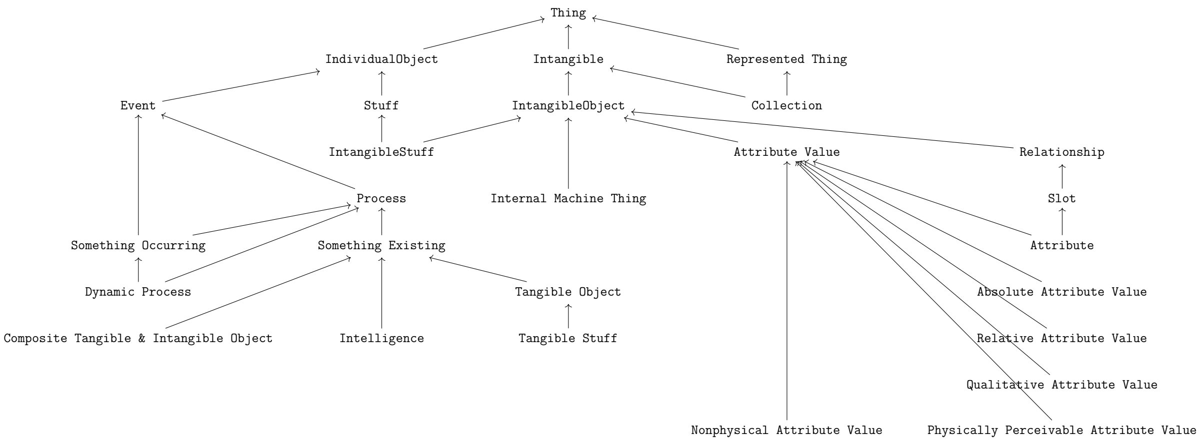


Figure 3.3: Cyc's top-level concept structure, adapted from [287, p. 84]. Observe how certain concepts inherit from more than one concept, such as `IntangibleStuff` and `Attribute`.

power [227]. Hence, the language is split between a frame language and a constraint language — the frame language encodes knowledge and the constraint language constrains knowledge and allows for reasoning. Additionally, the language is further split into so-called epistemological and heuristic levels, called EL and HL respectively, and most of the inference reasoning happens at the heuristic level because of efficiency concerns [226, p. 50-53].

What follows is an example of how to encode knowledge in CycL, creating effectively what is known as a *knowledge base*. Because CycL is built on top of Lisp, all CycL-specific keywords (called *units*) in the language are prefixed by `#%`, including constants, predicates, quantifiers, and functions. For instance, the aforementioned "is-a" predicate is represented in CycL as `#%IsA` (technically, predicates behave as truth functions). Suppose one would like to represent two employees of a certain company in CycL and specify both work in the same department. To do so, it is necessary to use CycL's frame and constraint languages.

First, one needs to declare two CycL units, `Alice` and `Bob`, which represent the two employees, as shown in listing 3.2.

```
#%Alice
    #%instanceOf         (#%ComputerEngineer #%Employee)
    #%age                (31)
    #%worksAt            (#%AcmeCompany)
    #%workingDepartment  (#%ResearchAndDevelopment)

#%Bob
    #%instanceOf         (#%Physicist #%Employee)
    #%age                (29)
    #%worksAt            (#%AcmeCompany)
    #%workingDepartment  (#%ResearchAndDevelopment)
```

Listing 3.1: Units representing knowledge about employees.

The knowledge base may be tested by declaring other units via the constraint language. Listing 3.2 illustrates some built-in units in CycL for knowledge reasoning and querying.

```
    (#%AllHaveSame (#%Employee #%allInstances) #%workingDepartment)
    (#%ThereExists x (#%Employee #%allInstances)
        (#%instanceOf x #%Physicist))
    (#%GreaterThan (#%Bob #%age) 30)
    (#%IsA #%Alice #%ComputerEngineer)
```

Listing 3.2: Statements on the knowledge base defined in figure 3.2. Observe the third statement is not true and hence if executed by Cyc, would return false.

BFO

The Basic Formal Ontology (BFO) is a top-level **o**ntology originally devised by Grenon and Smith [148] in the context of geospatial reasoning[3]. As opposed to Cyc, BFO does not aim for a comprehensive account of common-sense knowledge — rather, it is fully committed to being a small, upper **o**ntology designed for ontological integration.

BFO is heavily inspired by the Aristotelian approach to ontology building, and as such it is also clear on what philosophical principles it incorporates as ontological commitments [40]. At the top of the categories is `Entity`, similar to Cyc's `Thing`. All other categories descend from `Entity`, its direct children being `Continuant` and `Ocurrent`. `Occurrent` correspond to entities which persist through time by having temporal parts, such as a life or an action (smiling, frowning, raining), whereas `Continuant` correspond to entities existing in a given instant in time[4], such as a person or a smile. By making this distinction, Bihan and Barton [40] note BFO is recalling the philosophical issue of temporal parts, in particular the perdurantism vs. endurantism debate. BFO's answer to the debate is ecumenical, as it internalizes both views.

As a side effect of its ecumenical position, BFO may be broken down into two disjoint sub-**o**ntologies [249, p. 2]:

- SNAP: a series of *snap*shot **o**ntologies O_{ti} indexed by time — at the top of each **o**ntology lies the category `Continuant`
- SPAN: a single videoscopic **o**ntology O_v whose top category is `Occurrent`.

An **o**ntology in SNAP represents entities existing at a given moment in time, whereas SPAN represents entities unfolding through time. The SNAP **o**ntologies are further subdivided into 18 categories and SPAN is subdivided into 17 categories. Particularly, BFO also makes the distinction between universals and particulars [40] It is not in scope to dive into each of BFO's categories. However, the following may be asserted: unlike Cyc, BFO is intended to be light and extensible with only 36 categories as pictured in figures 3.4 and 3.5.

Otte, Beverley, and Ruttenberg [284] note BFO is committed, at a high-level, to three specific principles:

[3] As an interesting historical remark, it may be noted Pierre Grenon worked at Cycorp prior to developing BFO.

[4] It should be noted that such an entity may exist in *more than* one instant in time. However, its existence must be explicitly stated at all instants where applicable.

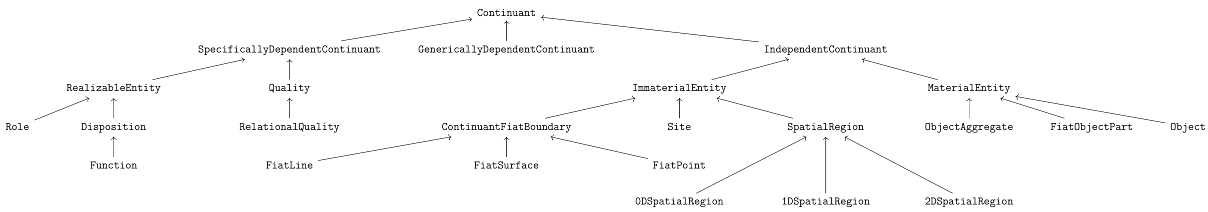


Figure 3.4: BFO categories from SNAP, extracted from [284].

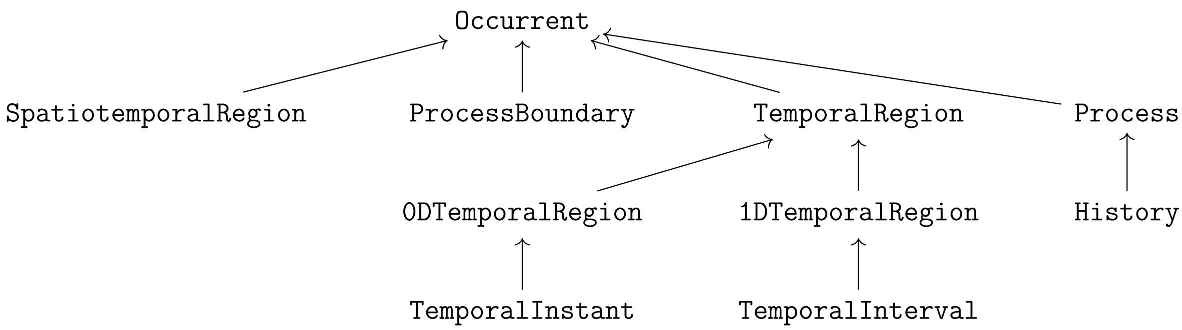


Figure 3.5: BFO categories from SPAN, extracted from [284].

1. Ontological Realism: BFO strives to represent actual reality and not language, concepts or mental representations of reality.

2. Fallibilism: BFO is subject to change given new scientific discoveries. This is evidenced by the fact that BFO has several versions with varying degrees of change among them.

3. Adequatism: In BFO, entities in scientific disciplines should not be reduced to some other domain of science deemed more fundamental.

As Otte, Beverley, and Ruttenberg [284, p. 5-7] note, BFO allows for mereological and spatio-temporal reasoning. Its theory of parts is based on Minimal Extension Mereology (MEM) [354] and its support for temporal reasoning is based on Allen's algebra [6]. Thus, the `is_a` relation is responsible for constructing the hierarchy of classes, and several other relations exist to encode and internalize MEM and Allen's algebra. For instance, an `Occurrent` may be a `temporal_part_of` another `Occurrent`; a `Continuant` may `participate_in` a `Process` (which in turn `is_a Ocurrent`). Unlike Cyc, BFO's `is_a` does not support multiple inheritance.

In terms of representation, BFO has been presented in several representation languages, depending on its *version*. Due to being rooted in fallibilism, there are actually several iterations or versions of BFO, the most distinct being BFO 1.1 and BFO 2.0 [343]. BFO 1.1 has been axiomatized using three different representation languages: a sub-language of the Web Ontology Language (OWL) named OWL-DL, the OBO Format (OBOF) and Isabelle, whereas BFO 2.0 dropped Isabelle in exchange for a dialect of Common Logic (CL) [366], the Common Logic Interchange Format (CLIF). Section 3.5 and chapter 7 discuss at length, but for the time being it suffices to note they are all based on (a fragment of) first-order logic. As a matter of fact, the ISO/IEC 21838-2:2021 standard axiomatizes BFO using first-order logic [191].

In order to gain a more a more concrete understanding of the nature of BFO, it is necessary to examine some fragments of BFO's OWL-DL and CLIF representation. Listing 3.2 presents the top-most category, `Entity`, in the OWL-DL representation language as extracted from the source code[5]. Listing 3.2 also presents the same top-most category, `Entity`, in CLIF.

```
<owl:Class rdf:about="http://purl.obolibrary.org/obo/BFO_0000001">
  <rdfs:label xml:lang="en">entity</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"
   />
  <ns2:BFO_0000179>entity</ns2:BFO_0000179>
  <ns2:BFO_0000180>Entity</ns2:BFO_0000180>
  <ns2:IAO_0000600 xml:lang="en">An entity is anything that exists or
    has existed or will exist. (axiom label in BFO2 Reference: [001
   001])</ns2:IAO_0000600>
  <ns2:IAO_0000112 xml:lang="en">Julius Caesar</ns2:IAO_0000112>
  <ns2:IAO_0000112 xml:lang="en">Verdi&apos;s Requiem</
   ns2:IAO_0000112>
  <ns2:IAO_0000112 xml:lang="en">the Second World War</
   ns2:IAO_0000112>
  <ns2:IAO_0000112 xml:lang="en">your body mass index</
   ns2:IAO_0000112>
  <rdfs:isDefinedBy rdf:resource="http://purl.obolibrary.org/obo/bfo.
   owl"/>
</owl:Class>
```

Listing 3.3: BFO's `Entity` represented in OWL-DL.

```
(cl-text http://ontohub.org/bfo/Entity.clif

  (cl:import http://ontohub.org/bfo/aux/unaryRelation.clif)
  (cl:import http://ontohub.org/bfo/exists_at.clif)

(cl:comment '**********************')
(cl:comment 'Declarations and defined expressions')
(cl:comment '**********************')

(unaryRelation Entity)

(cl:comment '**********************')
(cl:comment 'Axioms                ')
(cl:comment '**********************')

(forall (x)
  (if
    (Entity x)
      (exists (t)
      (and
        (TemporalRegion t)
        (exists_at x t)))))
```

[5]It should be noted the commentaries with examples of entities are indeed included in the source code

```
)
```

Listing 3.4: BFO's `Entity` represented in CLIF.

Observe that there aren't any axioms concerning `Entity`, as it is taken as a primitive. In OWL-DL, `Entity` is based on OWL's internal `Thing` class, while in CLIF it is represented via a unary relation. `TemporalRegion` is a category that has also been encoded in both representation languages. Listing 3.2 presents the `TemporalRegion` encoded in OWL-DL, whereas listing 3.2 presents the same category in CLIF.

```
<owl:Class rdf:about="http://purl.obolibrary.org/obo/BFO_0000003">
<rdfs:label xml:lang="en">occurrent</rdfs:label>
</owl:Class>
<!--  http://purl.obolibrary.org/obo/BFO_0000008  -->
  <owl:Class rdf:about="http://purl.obolibrary.org/obo/BFO_0000008">
  <rdfs:label xml:lang="en">temporal region</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://purl.obolibrary.org/obo/
   BFO_0000003"/>
  <owl:disjointWith rdf:resource="http://purl.obolibrary.org/obo/
   BFO_0000011"/>
  <owl:disjointWith rdf:resource="http://purl.obolibrary.org/obo/
   BFO_0000015"/>
  <owl:disjointWith rdf:resource="http://purl.obolibrary.org/obo/
   BFO_0000035"/>
  <ns3:IAO_0000602>(forall (x) (if (TemporalRegion x) (Occurrent x)))
    // axiom label in BFO2 CLIF: [100-001] </ns3:IAO_0000602>
  <ns3:IAO_0000601 xml:lang="en">Every temporal region t is such that
    t occupies_temporal_region t. (axiom label in BFO2 Reference:
   [119-  002])</ns3:IAO_0000601>
  <ns3:IAO_0000116 xml:lang="en">Temporal region doesn&apos;t have a
   closure axiom because the subclasses don&apos;t exhaust all
   possibilites. An   example would be the mereological sum of a
   temporal instant and a temporal interval that doesn&apos;t overlap
    the instant. In this case the   resultant temporal region is
   neither 0-dimensional nor 1-dimensional</ns3:IAO_0000116>
  <ns3:BFO_0000180>TemporalRegion</ns3:BFO_0000180>
  <ns3:BFO_0000179>t-region</ns3:BFO_0000179>
  <ns3:IAO_0000602>(forall (r) (if (TemporalRegion r) (
   occupiesTemporalRegion r r))) // axiom label in BFO2 CLIF:
   [119-002]   </ns3:IAO_0000602>
  <ns3:IAO_0000602>(forall (x y) (if (and (TemporalRegion x) (
   occurrentPartOf y x)) (TemporalRegion y))) // axiom label in BFO2
   CLIF: [101-  001] </ns3:IAO_0000602>
  <ns3:IAO_0000600 xml:lang="en">A temporal region is an occurrent
   entity that is part of time as defined relative to some reference
   frame.   (axiom label in BFO2 Reference: [100-001])</
   ns3:IAO_0000600>
  <ns3:IAO_0000601 xml:lang="en">All parts of temporal regions are
   temporal regions. (axiom label in BFO2 Reference: [101-001])  </
   ns3:IAO_0000601>
<rdfs:isDefinedBy rdf:resource="http://purl.obolibrary.org/obo/bfo.
   owl"/>
</owl:Class>
```

```
  <!--  http://purl.obolibrary.org/obo/BFO_0000011  -->
  <owl:Class rdf:about="http://purl.obolibrary.org/obo/BFO_0000011">
  <rdfs:label xml:lang="en">spatiotemporal region</rdfs:label>
</owl:Class>
  <!--  http://purl.obolibrary.org/obo/BFO_0000015  -->
  <owl:Class rdf:about="http://purl.obolibrary.org/obo/BFO_0000015">
  <rdfs:label xml:lang="en">process</rdfs:label>
</owl:Class>
  <!--  http://purl.obolibrary.org/obo/BFO_0000035  -->
  <owl:Class rdf:about="http://purl.obolibrary.org/obo/BFO_0000035">
  <rdfs:label xml:lang="en">process boundary</rdfs:label>
</owl:Class>
```

Listing 3.5: TemporalRegion represented in OWL-DL.

```
(cl:text http://ontohub.org/bfo/TemporalRegion.clif

  (cl:import http://ontohub.org/bfo/Occurrent.clif)

(cl:comment '**********************')
(cl:comment 'Declarations and defined expressions')
(cl:comment '**********************')


(Universal TemporalRegion)
  (isA TemporalRegion Occurrent)

(forall (x)
 (iff
  (TemporalRegion x)
  (occurrent_part_of x time_R)
  ))

(cl:comment 'Because of the reflexivity of occurrent_part_of it
   follows:
(TemporalRegion time_R)
')

(cl:comment '**********************')
(cl:comment 'Axioms                ')
(cl:comment '**********************')

(cl:comment "All temporal regions are either 0D or 1D. [FN009-001]")
(forall (t)
  (iff
  (TemporalRegion t)
  (or
   (1DTemporalRegion t)
   (0DTemporalRegion t))))

(cl:comment "Every temporal region occupies_temporal_region itself.
   [137-001]")
```

```
(forall (x)
  (if
    (TemporalRegion  x)
  (occupies_temporal_region x x)))

(cl:comment '*********************')
(cl:comment 'Theorems                ')
(cl:comment '*********************')

(cl:comment "All occurrent_parts of temporal regions are temporal
   regions. [101-002]")

(forall (x y)
  (if
   (and
   (occurrent_part_of x y )
    (TemporalRegion x))
  (TemporalRegion y)  ))
)
```

Listing 3.6: `TemporalRegion` represented in CLIF.

Despite both languages being deemed adequate to represent BFO by its developers, they are quite distinct not only in terms of syntax but also modularity, orthogonality, expressivity, implementation details, and performance. As BFO is an open source ontology[6], the interested reader may refer to the actual code to understand its inner workings.

DOLCE

The Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) is a top-level ontology initially developed as part of the WonderWeb Project [250] and remains under active development at the Istituto di Scienze e Tecnologie della Cognizione (CNR-ITSC)'s Laboratory of Applied Ontology led by Guarino [249, p. 4]. Borgo et al. [44] present a historical account on DOLCE and its variants, which will be reproduced shortly.

Similarly to other top-level ontologies, DOLCE is clear on what principles it stands. DOLCE is, as its name suggests, designed towards human common sense and natural language, and it is committed to ease interaction with human agents by "making already formed conceptualizations explicit" [40, 250], positioning itself fully as descriptive. As Borgo et al. [44, p. 2] note, DOLCE's response to the universal vs. particular debate is that all entities are particulars, whereas properties and relations are universals. This means the top-level class in DOLCE's taxonomy is explicitly `Particular`, as depicted in figure 3.6. DOLCE sits in-between BFO and Cyc in terms of dimensions, with around 200 classes and 40 axioms in its OWL representation.

In [154], Guarino outlines a personal history of DOLCE and compares it with BFO in terms of principles and philosophical choices. Borgo et al. [44, p. 3-4] clearly lay out the principles behind DOLCE's taxonomical and axiomatic choices:

- Continuants vs. occurrents: Similarly to BFO, DOLCE distinguishes and internal-

[6] https://github.com/BFO-ontology/BFO

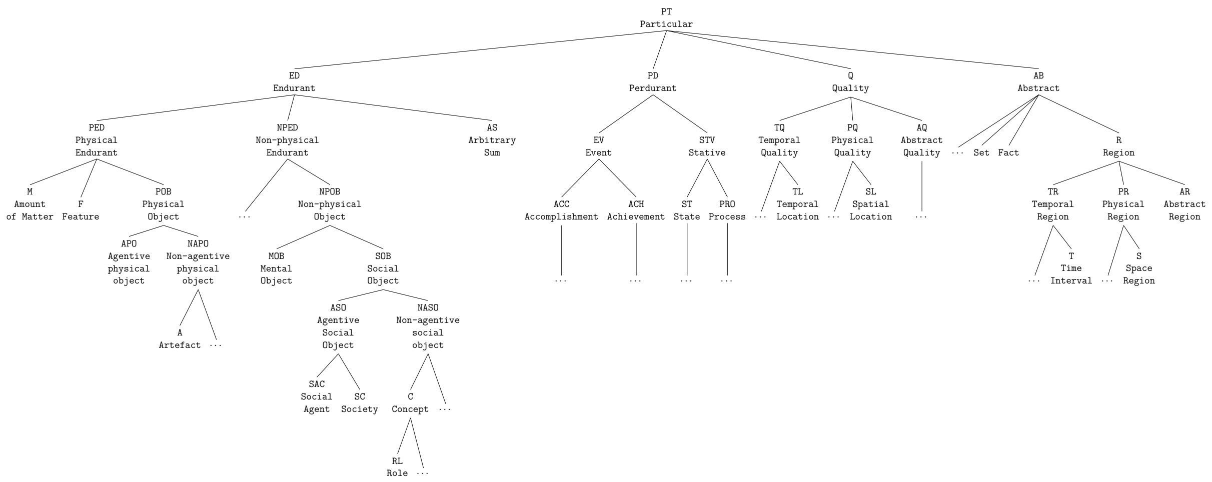


Figure 3.6: Fragment of DOLCE's taxonomy, extracted from [44].

izes endurants (in BFO's terminology, ocurrents) and perdurants (in BFO, continuants). In order to connect these two kinds of entities, DOLCE provides a special relation called "participation" — an endurant *participates* in a perdurant. This is radically different compared to BFO, where continuants and ocurrents (i.e. the SNAP and SPAN **o**ntologies) are seen as "different views" of reality, which do not connect or talk with each other [154, p. 7].

- Independent vs. dependent entities: DOLCE also distinguishes between dependent and independent entities. Dependent entities, such as features, depend on some physical object to exist, which in turn depend on the existence of at least one event (its life).

- Processes vs. events: As per figure 3.6, processes and events are different types of perdurants. Their distinction is mereological — in short, events are not cumulative.

- Properties, qualities and quantities: In DOLCE, qualities are entities which can be perceived and measured. DOLCE includes a typology of qualities to allow for type-based comparison of qualities.

- Function and role: Despite having a `Role` class, DOLCE does not explicitly formalize functions and roles (unlike BFO) but includes a "classification" relation which connects roles to other entities.

- Relations: Aside from aforementioned "participation" and "classification" relations, DOLCE also contains "parthood" and "constitution" relations. "Parthood" is time-indexed when connecting endurants and atemporal when connecting perdurants or abstracts, while "constitution" relates spatio-temporally co-located entities. As an example, Borgo et al. [44] cite the relation between the matter constituting a statue and the statue itself.

Concerning representation languages, DOLCE was the first top-level **o**ntology to be formalized using first-order logic [250], with a proof of consistency later devised by Kutz and Mossakowski [212]. The actual logic of choice was the first-order quantified modal logic **QS5** as presented by Fitting and Mendelsohn [121] plus the Barcan and converse

Barcan formulas which state, when interpreting modality as necessity and possibility, that all *possible* entities are *real* entities. This characterizes DOLCE as a *possibilist* ontology, i.e., its domain of discourse consists of all possible entities [44, p. 5].

The axioms below are extracted from DOLCE's **QS5** axiomatization and relate to the "constitution" relation, as presented in [250] and later re-presented in [44]. The right-side notes refer to the axiom labels in [250].

$$
\begin{array}{lr}
\mathrm{K}(x,y,t) \rightarrow ((\mathtt{ED}(x) \vee \mathtt{PD}(x))) \wedge (\mathtt{ED}(y) \vee \mathtt{PD}(y)) \wedge \mathtt{T}(t) & \text{Constitution typing, Ad20} \\
\mathrm{K}(x,y,t) \rightarrow (\mathtt{PED}(x) \longleftrightarrow \mathtt{PED}(y)) & \text{Ad21} \\
\mathrm{K}(x,y,t) \rightarrow (\mathtt{NPED}(x) \longleftrightarrow \mathtt{NPED}(y)) & \text{Ad22} \\
\mathrm{K}(x,y,t) \rightarrow (\mathtt{PD}(x) \longleftrightarrow \mathtt{PD}(y)) & \text{Ad23} \\
\mathrm{K}(x,y,t) \rightarrow \neg \mathrm{K}(y,x,t) & \text{Ad24} \\
(\mathrm{K}(x,y,t) \wedge \mathrm{K}(y,z,t)) \rightarrow \mathrm{K}(x,z,t) & \text{Ad25} \\
\mathrm{K}(x,y,t) \rightarrow (\mathrm{PRE}(x,t) \wedge \mathrm{PRE}(y,t)) & \text{Ad26} \\
\mathrm{K}(x,y,t) \longleftrightarrow \forall t'(\mathrm{P}(t',t) \rightarrow \mathrm{K}(x,y,t')) & \text{Ad27}
\end{array}
$$

As previously mentioned, "constitution" is a temporal relation between two entities, it is represented as $\mathrm{K}(x,y,t)$ meaning x constitutes y at instant in time t. The symbol t' represents an instant in time different from t. The symbols PRE and P represent the relations "being present at" and "parthood" — $\mathrm{PRE}(x,t)$ means x is present at t and $\mathrm{P}(x,y)$ means x is part of y while $\mathrm{P}(x,y,t)$ means x is part of y at time t[7]. The terms $\mathtt{XYZ}(x)$ reads as x is `XYZ`, where `XYZ` is one of DOLCE's classes (e.g. `ED`, `PD`, `T`, etc.) as shown in figure 3.6.

As **QS5** is not a representation language, nor is it decidable [208], Masolo et al. [250] also represented DOLCE in the Knowledge Interchange Format (KIF) [132], a predecessor of CLIF. DOLCE also has different versions, but unlike BFO it is not versioned in a sequential sense. DOLCE's versions, such as DOLCE-lite, DOLCE-ultralite, DOLCE-zero, are actually application-oriented variants with smaller taxonomies and reduced sets of axioms [176, 292]. These variants have been represented in OWL, the Resource Description Framework (RDF) and DAML+OIL. Listing 3.2 illustrates how DOLCE-Ultralite represents the `Concept` class in RD.

```
    :Concept
    a owl:Class ;
    rdfs:comment """A Concept is a SocialObject, and isDefinedIn some
    Description; once defined, a Concept can be used in other
   Description(s). If a Concept isDefinedIn exactly one Description,
   see the LocalConcept class.
The classifies relation relates Concept(s) to Entity(s) at some
   TimeInterval"""^^xsd:string ;
    rdfs:isDefinedBy <http://www.ontologydesignpatterns.org/ont/dul/
   DUL.owl> ;
    rdfs:label "Concept"@en, "Concetto"@it ;
```

[7] Notice that DOLCE overloads its language in a computational sense. The same symbol, P, technically represents two different relations depending on arity.

```
    rdfs:subClassOf :SocialObject, [
        a owl:Restriction ;
        owl:onProperty :isDefinedIn ;
        owl:someValuesFrom :Description
    ], [
        a owl:Restriction ;
        owl:allValuesFrom :Concept ;
        owl:onProperty :hasPart
    ] ;
    owl:disjointWith :InformationObject, :Situation, :SocialAgent .
```

DOLCE and all of its variants are open-source. The initial **QS5** representation of DOLCE can be found in the original WonderWeb report [250], alongside its KIF representation.

SUMO

The Suggested Upper-Merged Ontology (SUMO) is the largest openly-available top-level **o**ntology, with more than 25.000 categories or classes (called "terms" in SUMO) and 80.000 axioms. It was initially presented by Niles and Pease [278], developed in the context of the Standard Upper Ontology Working Group. Similarly to Cyc, SUMO is on the edge of what can be considered a top-level **o**ntology, for its categories range from the top-level class `Entity` down to `Game`, `Surgery` and `WarmBloodedVertebrate` [295].

SUMO is particular in that it is very openly a "modular **o**ntology", much more so than BFO, possessing many sub-**o**ntologies. In fact, the upper-level portion of SUMO is regarded by its authors as a sub-**o**ntology in itself, and it is further broken down into other sub-**o**ntologies that exhibit different dependencies among each other. There is also a mid-level sub-**o**ntology inside of SUMO called MId-Level Ontology (MILO) whose goal is to connect the upper-level SUMO to its 30 currently available domain sub-**o**ntologies. Thus, the entirety of SUMO consists of the upper-level sub-**o**ntology, MILO and all the domain sub-**o**ntologies.

Figure 3.7 depicts the taxonomy of SUMO's top-level sub-**o**ntology. Although this is not the case for the top-level portion, SUMO allows for multiple taxonomical inheritance, akin to Cyc.

SUMO is represented in SUO-KIF, a simplified variant of the KIF representation language. Listing 3.2 is an excerpt of the `HouseHoldAppliances` sub-**o**ntology documenting an axiom concerning the entity `BodyCareAppliance`:

```
;; BodyCareAppliance
(subclass BodyCareAppliance HouseholdAppliance)

(documentation BodyCareAppliance EnglishLanguage "&%
 HouseholdAppliance
designed to be used for exercise or personal body care.")

(termFormat EnglishLanguage BodyCareAppliance "body care appliance
 ")

```

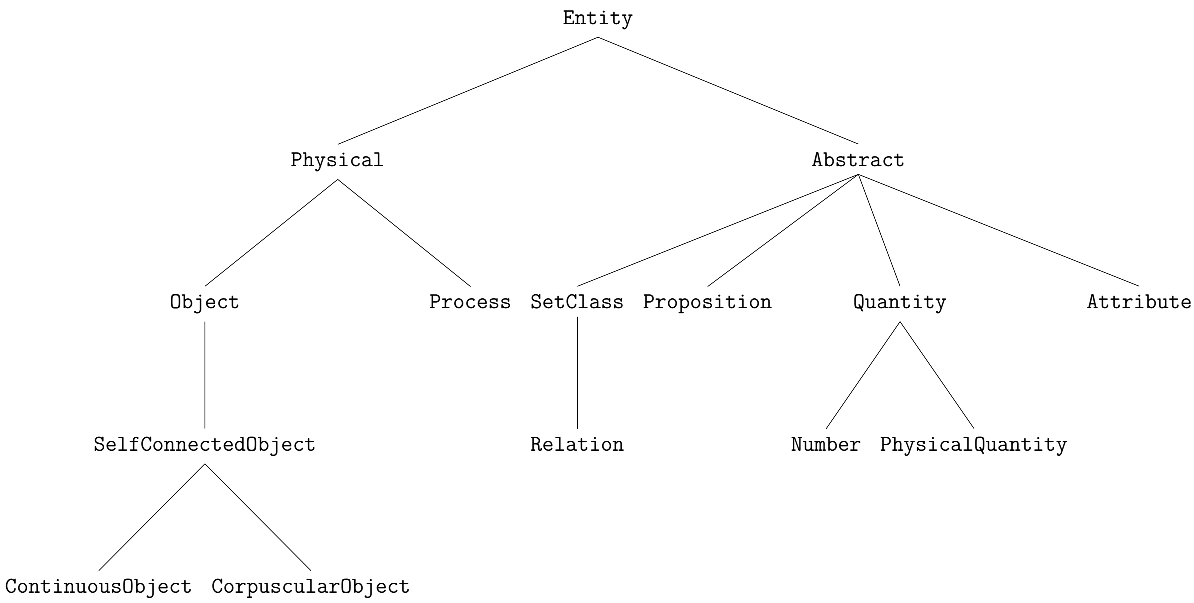


Figure 3.7: SUMO's top level classes.

```
; if it is BodyCareAppliance then hasPurpose - patient is Human/
 BodyPart
(=>
  (instance ?DEVICE BodyCareAppliance)
  (hasPurpose ?DEVICE
    (exists (?PROC ?PATIENT)
      (and
        (instance ?PROC Process)
        (instrument ?PROC ?DEVICE)
        (patient ?PROC ?PATIENT)
        (or
          (instance ?PATIENT Human)
          (instance ?PATIENT BodyPart))))))
```

In contrast to BFO and DOLCE, the philosophical assumptions underlying SUMO are rarely discussed or brought to the fore. Magee [240, p. 246] remarks this may be due to SUMO's own syncretic nature. SUMO was constructed by manually merging several prior ontologies into a single hybrid ontology. As a result, it inherits the majority of the underlying assumptions of its constituent ontologies. Nevertheless, as evidenced by Niles and Pease [278] themselves, achieving a harmonious integration was not always feasible, necessitating some difficult decisions. For instance, for instance, SUMO adopts an endurantist perspective.

SUMO is also an open source ontology[8] and enjoys specific tools for developing and using it, such as the SUMOjEdit text editor [294] and the SigmaKEE integrated development environment [296].

[8]https://github.com/ontologyportal/sumo/

UFO

The Unified Foundational Ontology (UFO) was presented by Guizzardi [165] in his PhD thesis, being initially developed over the course of several papers [166]. Guizzardi proposed UFO as a new foundational **o**ntology explicitly geared towards conceptual modeling, rooted in several philosophical and non-philosophical domains, such as cognitive science and linguistics. As noted by Guizzardi et al. [167], UFO's overall goal is to ease conceptual modelling:

> The ultimate goal of this project is, thus, providing well-founded engineering mechanisms for helping modelers to achieve *intra-worldview consistency*, i.e., ontological consistency when taking the world a certain way, and *inter-worldview interoperability*, i.e., making explicit the ontological commitments of a worldview such that different worldviews can safely interoperate [167, p. 4]

UFO is structured as a four-category **o**ntology in the sense of [233]. A four-category **o**ntology contains four fundamental categories of beings: substantial and non-substantial particulars; substantial and non-substantial universals. UFO contains several micro-theories, each representing a fundamental conceptual modelling notion stemming from ontological foundations [167]. Some of UFO's micro-theories include:

- Theory of types and taxonomic structures: this theory is concerned with the categorization and hierarchy of objects, termed object identifiers, and is formalized in a sortal quantified modal logic;
- Theory of part-whole relations: similarly to other top-level **o**ntologies, UFO contains the requisite conceptual framework for describing mereological relations;
- Theory of particularized intrinsic properties, attributes and attribute value spaces: this theory is concerned with properties unique to particular entities;
- Theory of particularized relational properties and relations: this encodes the properties and relations between entities and proposes a way to link particularized properties to propositions via Weak Truthmaking [161]
- Theory of roles: this theory is concerned with the roles entities can embody and what identities they might take on;
- Theory of events: this is a spatiotemporal theory with mereological aspects, it states how events may be ordered and how they relate to causation and change;
- Theory for multi-level modeling: this theory provides a framework for hierarchical or layered structure descriptions;

Furthermore, UFO distinguishes entities between endurants and perdurants. In a manner analogous to BFO, UFO has sub-**o**ntologies for addressing both kinds of entities and an additional one for a third kind of entities:

- **UFO-A**, an **o**ntology of endurants, i.e. entities which do not change in time;

- **UFO-B**, an **o**ntology of perdurants, i.e. entities which unfold in time and have temporal parts, referred to as events in UFO;
- **UFO-C**, an **o**ntology of social and intentional entities built on the foundations of **UFO-A** and **UFO-B**.

In order to provide an overview of the UFO taxonomy and to introduce some of the terminology used, a *kind* is a sort of endurant type, this is a rigid type which allows instantiation of individuals in all applicable situations. Examples of kinds include dog, car, book, organization. Intuitively, kinds refer to concepts described by substantives. In fact, kinds or specializations of kinds are *rigid sortals*. Sortals need not necessarily be rigid, for they may also be *anti-rigid* and further divided into *phase* and *roles*. Roles encode the relational classification conditions of contigent nature, such as musician as a role of a person in the scope of a band relator, employee in the role of a person in the scope of an employment relator.

Figure 3.8 presents the taxonomy of UFO classes, as extracted from [167], illustrating how kinds, sortals and roles relate.

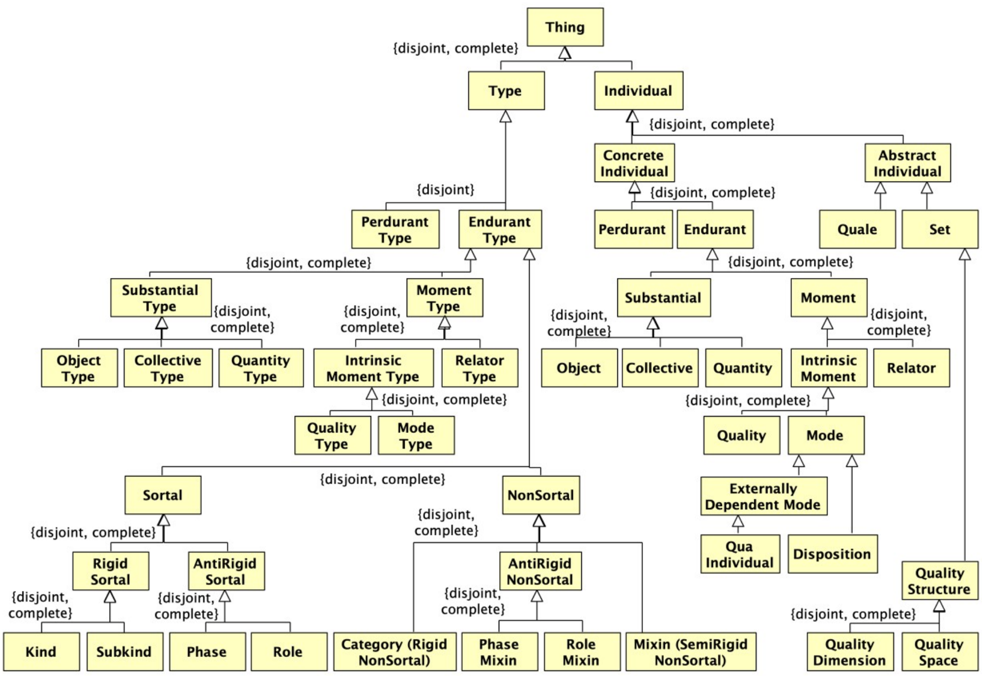


Figure 3.8: A fragment of UFO's taxonomy, reproduced from [167, p. 5].

In terms of representation language, UFO has been fully formalized using first-order logic in [305][9]. Guizzardi et al. [167] formalized UFO's micro-theory of types and taxonomic using **QS5**, similarly to DOLCE. A distinguinshing feature of UFO is its utilization in the development of an **o**ntology-driven conceptual modelling language, OntoUML, which in turn is now used to represent UFO-backed models [165]. OntoUML, very much like its counterpart UML (which stands for Unified Modeling Language), has a visual representation through diagrams. Figure 3.9 depicts how the sentence "There is a four-legged table made of wood. Some time later, a leg of the table is replaced. Even later, the table

[9]Openly available at https://github.com/unibz-core/ufo-formalization

is demolished so it ceases to exist although the wood is still there after the demolition." may be represented in OntoUML, also extracted from [167, p. 18].

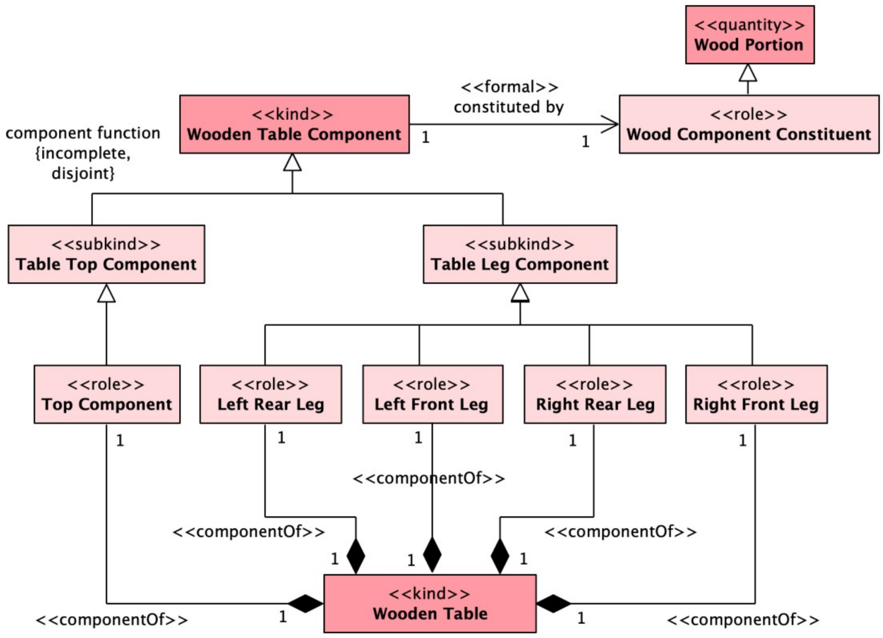


Figure 3.9: Sample of knowledge representation in OntoUML, from [167, p. 18].

Current research on UFO is mainly carried out by the Ontology and Conceptual Modeling Research Group (NEMO), based in the Federal University of Espírito Santo. An implementation of a subset of UFO, developed by NEMO, is open source and implemented in RDF for OWL compatibility[10].

And many others

It should be noted that there are a number of top-level ontologies that are currently in using. Some further examples are the General Formal Ontology (GFO) [231], the Generalized Upper Model (GUM) [26], the TUpper Ontology [152], Yet Another More Advanced Top-leve Ontology (YAMATO) [258], the Object-Centered High-Level Reference Ontology (OCHRE) [340]. There is indeed an entire issue from the *Applied Ontology* journal dedicated to top-level ontologies [45].

Two key points pertaining to top-level ontologies will be highlighted:

- There is diversity in unity: despite the fact that all top-level ontologies share a common goal, namely to create a common formal representation of reality in a practical sense, there is much diversity not only in terms of underlying philosophical choices but also in pragmatic decisions such as representation language and size.
- Committing to a particular top-level ontology can be costly: this may be perceived as a consequence of the preceding point. Due to the high degree of diversity, fully committing to one specific ontology to develop a domain or task ontology can be a time-consuming endeavor.

[10] Available at https://nemo-ufes.github.io/gufo/

3.3 Domain and task ontologies

This section presents a few examples of **o**ntologies based on previously discussed top-level ontologies. This overview will be brief and will focus on key points, as discussing each domain and task **o**ntology can rapidly evolve into a dedicated discussion. Additionally, it is necessary to recall the general objective of this chapter is to not focus on any particular **o**ntology.

- Combined Security **O**ntology (CSO) [1]: CSO is an a domain **o**ntology based on UFO whose goal is to align terminology for the security and safety domains.
- Goal Oriented **O**ntology (GORO) [271]: GORO is task **o**ntology also based on UFO whose purpose is to formalize concepts of Goal-Oriented Requirements Engineering (GORE).
- ROMAIN [204]: ROMAIN is a domain **o**ntology based on BFO whose purpose is to represent industrial maintenance knowledge and relationships.
- Domain **o**ntology for offshore petroleum production plants (O3PO) [334]: O3PO is another domain **o**ntology based on BFO to deal with semantic interoperability issues in oil and gas. It has also been validated using real-world data from Brazil's Mero oil field.
- Food processing **o**ntology (Onto-FP) [268]: Onto-FP is a task **o**ntology based on DOLCE geared towards representing food product transformations with a temporal aspect. In [268], the authors detail an application in the domain of winemaking.
- Enterprise Application Integration Ontology (ENIO) [48]: ENIO is a domain **o**ntology based on both DOLCE and SUMO via alignment and is concerned with the domain of enterprise application integration.

These few examples highlight how far and wide top-level **o**ntologies may spread in various domains and tasks. As a last remark, the reader may refer to De Baas et al. [110]'s review of 40 domain **o**ntologies only in the domain of Materials Science alone.

3.4 Deviant ontologies

In conclusion to the examination of **o**ntologies, it is necessary to consider a specific deviant **o**ntology that stands out: the Gene Ontology (**GO**) [91].

GO may be considered the most successful **o**ntology to date, despite not being based on any top-level **o**ntology. In fact, **GO** was developed prior to the development, let alone the implementation, of any top-level **o**ntology [14]. It is a domain **o**ntology which describes biological knowledge through three disjoint application sub-**o**ntologies:

- Molecular Function: a sub-**o**ntology describing molecular-level activities performed by gene products;

- Cellular Component: a sub-**o**ntology describing cellular structures where a gene performs a function;

- Biological Process: a sub-**o**ntology describing organism-level activities accomplished by several molecular functions.

In terms of size, as of 2024 **GO** contains over 42 thousand terms and over 7 million gene annotations, covering over 5 thousand different species. Hence, despite its non-conformity to Guarino's taxonomy, **GO** may be considered a massive **o**ntology.

Unlike the aforementioned top-level **o**ntologies, **GO** has no clear statements of its ontological commitments. However, **GO** does exhibit a sort of prototypical notion of mereology as can be seen by its four kinds of relations between entities: *is a*, *part of*, *has part* and *regulates*.

As far as representation languages, **GO** is described in OWL and also in OBOF. Despite not being based on any prior **o**ntology, **GO** is part of the OBO foundry [194].

The success of **GO** highlights that deviant **o**ntologies should be taken into consideration when proposing tools for ontological interoperation. Another widely used deviant **o**ntology, whose discussion is not in scope for this section, is the Foundational Model of Anatomy (FMA)[11].

3.5 Developing ontologies

As this chapter draws to a close, it is imperative to briefly examine the underlying methods, tools and ways of working to develop **o**ntologies. The past sections presented a few representation languages for **o**ntologies. However, the matter of choosing and developing representation languages is only a fragment of a whole area of study named **o**ntology engineering. **O**ntology engineering is concerned with the overall methods and methodologies for building presentations.

Corcho, Fernández-López, and Gómez-Pérez [94] present an early overview of methods, tools and languages for building **o**ntologies, the main one of which being OntoClean, a meta-theoretical framework for **o**ntology development. OntoClean was developed by Guarino and Welty [163] as the first attempt to formalize ontological analysis in a systematic way. It is based on establishing four metaproperties of **o**ntologies: identity, unity, rigidity and dependence. Later on, in OntoClean 2.0, two more metaproperties were added: permanence and actuality [396]. The point of OntoClean is to use these metaproperties to assess and compare key traits of **o**ntologies.

Some other methodological frameworks for building ontologies are the Karlsruhe **o**ntology (KAON) [239] and the Developing Ontology-Grounded Methods and Applications (DOGMA) framework [370]. Both frameworks have been developed to manage manual **o**ntology development.

In recent times, there has been a notable shift in the focus of the literature towards the task of (semi)-automated **o**ntology learning from data. **O**ntology learning has always been an area of interest in **o**ntology engineering — for instance, Buitelaar, Olejnik, and

[11]Openly available at http://sig.biostr.washington.edu/projects/fm/AboutFM.html/.

Sintek [57] present an extension of the **o**ntology editor Protégé [134] with this purpose. However, more recent advances in natural language processing and machine learning have permitted the development of more sophisticated methods for **o**ntology learning, such as those described in [15, 397]. It appears that automated **o**ntology learning and evaluation may continue to be a significant area of focus in **o**ntology engineering in the near future.

Chapter 4

The Meta-Ontological Problem

THIS chapter introduces and defines a problem within Meta-Ontology that has been discussed throughout the previous chapters. It will examine how Applied Ontology deals with this problem by investigating its philosophy and ultimately take a stance on the problem.

4.1 Definition and Motivation

AS a first step, it is necessary to clearly define the key problem that will be addressed in this chapter, which bears the same name as the chapter itself.

Definition 4.1.1 (Meta-Ontological Problem)**.** The Meta-Ontological Problem (MOP) may be informally defined as the following two-part questions:

1. What does it mean for an ontological account to be *correct*?
2. How many correct ontological accounts are there if any?

In the literature of applied Ontology, Guarino [158] was one of the first to analyze a ramification or a particular aspect of the MOP by looking at the so-called *interaction problem* coined by Bylander and Chandrasekaran [58]:

> Representing knowledge for the purpose of solving some problem is strongly affected by the nature of the problem and the inference strategy to be applied to the problem [58, p. 232].

With regard to the interaction problem, Guarino [158] sustains the thesis the domain knowledge should be independent of a particular problem. The domain knowledge, according to Guarino, should be modelled so as to enable reusability. This observation led to his taxonomy, presented in chapter 3, and it is indirectly the intuitive basis for description logics, presented in chapter 5.

Nevertheless, it appears that Guarino's thesis is insufficient to address the MOP. This is because the MOP is not only concerned with a particular issue of representing knowledge of a given domain or solving a particular problem. In fact, its roots in the broader context of philosophy of science can be traced to the work of Duhem [116] on the underdetermination of scientific theories in physics and, perhaps unsurprisingly, to the work of W. V. Quine [314].

Recalling chapter 2, W. V. Quine [314] argues an ontological account is a reflection of the ontological commitments one makes with respect to a theory. Furthermore, he argues that empirical observation does not validate a theory — what it does is, perhaps, validate whether *all* ontological commitments of such theory made are consistent or not with the empirical observations that have been made. This implies that it is not possible to assess consistency of a single ontological commitment with respect to an empirical observation.

Going further back in time, prior to Quine's work on the philosophy of science, Descartes [112] had already written about prototypical ramifications of the MOP. Descartes [112] presents different "demons" whose purpose is to distort empirical observations and to impede any kind of possibility of assessing the "correctness" of any ontological account. Neither Descartes, nor W. V. O. Quine, nor Duhem have clearly stated the MOP, however they all touched on assumptions, causes, consequences or limitations of the MOP, suggesting its fundamental status in philosophical discourse.

4.2 A potential reduction

In the spirit of complexity theory, it is conceivable that the MOP might be reducible to a language problem if it were to be limited to the domain of applied Ontology. The term language does not correspond to the discipline, in the Wittgensteinian sense — rather, the problem at hand is the problem of choosing a language **L** to build an **o**ntology. This section posits that a reduction already permeates the literature of applied Ontology, as will be demonstrated clear shortly. Additionally, this section will attempt to elucidate why such reduction is at times taken for granted and why so.

As a first step, we define a sub-problem of the MOP for applied Ontology:

Definition 4.2.1 (Applied Meta-Ontological Problem)**.** The Applied Meta-Ontological Problem (AMOP) may be informally defined as the following two-part questions:

1. What does it mean for an **o**ntology to be *correct*?
2. How many correct **o**ntologies are there if any?

As chapter 2 discussed, an **o**ntology must, by definition, be described by a formal language **L**. Furthermore, chapter 3 illustrated, virtually all existing **o**ntologies have been built on representation languages either in part or fully based on *logics*. In fact, Guarino's definition (definition 2.2.12 in chapter 3) takes for granted that **L** is a logic of some sort and ontological commitments are represented via an axiomatization of the domain, viz. through a set of formulas comprising a logical theory. It is evident that the task of establishing the axiomatization lies at the meta-discussion level[1]. Nevertheless, there is an underlying task to be addressed: how does one go about choosing which **L** or, more specifically, which logic to use?

[1]As the reader may recall, an **o**ntology is a *shared* conceptualization

The past observations may be summarized as the following problem in the discipline of Logic:

Definition 4.2.2 (Meta-Logical Problem)**.** The Meta-Logical Problem (MLP) may be informally defined as the following two-part questions:

1. What does it mean for a logic to be *correct*?
2. How many correct logics are there if any?

The preceding observations on the fact that, in applied Ontology, an **o**ntology depends on a logic by definition, lead to the following thesis:

Thesis 4.1 (AMOP-MLP Reduction Thesis)**.** The AMOP is partially reducible to the MLP via a linguistic argument. In other words, solving the MLP is one of the steps required to solve the AMOP.

One might inquire as to how plausible the thesis is. By looking back at chapter 2, it can be argued that the thesis has been assumed throughout much of the existing literature. To begin with, when initially presenting his definition of **o**ntology, Gruber [150, p. 909] remarked that "formally, an ontology is the statement of a logical theory", which Kutz and Mossakowski [212, p. 259] interpret as follows:

> [...] on a technical level, an ontology is seen as equivalent to a logical theory, written in a certain formalism. [212, p. 259]

It thus appears that the ontological game was rigged from the outset. This is further evidenced when, for instance, Lange et al. [216, p. 2] states that first-order logic is *required* for formalizing mereology and spatio-temporal reasoning — both of which are clearly ontological concepts. Finally, it is common practice in applied Ontology to systematically analyze and compare representation languages (which are based on logics) in order to choose one to build out an **o**ntology (see, for instance, [95, 402]).

4.2.1 Side effects of reducing

Most of the literature seemingly assumes the thesis in varying degrees of explicitness. Nevertheless, it is still of interest to underscore some of the side effects that have been introduced by it, either directly or indirectly.

If the thesis is assumed to be true, the task of determining the correctness of an **o**ntology according to some definition depends on the task of determining the correctness of a logic according to some definition further parameterized by a definition of ontological correctness. One may build two different **o**ntologies $\mathbf{O}_1$ and $\mathbf{O}_2$ using the same language (logic) $\mathbf{L}$, but the question of how well each **o**ntology represents their domain, according to the reduction thesis, is not only a matter of what ontological axioms are established and how well they are written — it is initially a matter of how precisely the underlying logic $\mathbf{L}$ itself is able to capture the intended ontological content of an axiom.

Some authors are against the reduction thesis 4.1, such as Neuhaus and Hastings [275] for practice-based reasons. Indeed, reducing the task of developing or choosing an

ontology to the task of building or developing a logic is at odds with the shared, interoperable aspect of **o**ntology's own definition. Neuhaus and Hastings's main argument is **o**ntology building inherently requires consensus amongst the **o**ntology users and developers, since "a pre-existing 'shared conceptualisation' is a rare luxury" [275, p. 20] and adding logical axioms to a theory is merely a step in the process.

Additionally, the thesis can only be taken as true if one agrees that Logic as a discipline and logics as the subject of such discipline are devoid of ontological content. For if Logic has ontological content of its own, the ontology of Logic could be reduced by Logic itself as per the thesis. This would entail the reduction thesis to be circular in a particular case and thus either false or trivially true (i.e. if a "no-operation" reduction is acceptable). This actually corresponds to what Cocchiarella [85] referred to as the logic as language vs. logic as calculus debate, recalling the work of Heijenoort [175], which in turn echoes the syntax vs. semantics divide in Logic as a discipline. The view of logic as language imbues logics with ontological content (embedded semantics for a certain domain), while the view of logic as calculus takes logics to be abstract calculi which allow for many interpretations over varying domains. In practice, though, this debate is not a real threat to the thesis — as Cocchiarella [85, p. 8] notes, the current consensus not only in applied Ontology but in Philosophy as a whole is to see logic as calculus.

4.3 How to solve the problem?

FROM this point onward, the reduction thesis 4.1 shall be taken as granted and the focus shall shift towards the MLP. In essence, there are two possible approaches ways to address the problem: assuming logical monism, which entails the pursuit of the "one true logic", *or* the assumption of logical pluralism, which necessitates the ability to manage a plethora of logics.

What follows is a sketch of an argument for logical pluralism. The reason it is a sketch is two-fold. Firstly, the argument is not concerned with the epistemological and metaphysical implications of choosing one position over the other — in fact, this section does not make mention of or considers if there are any. Secondly, the argument is very much pragmatic and observational. It is a fact that there are already a plethora of different **o**ntologies at varying levels of domain with different representation languages, and as a consequence there are several underlying logics of choice.

In many scenarios, it is either desired or simply necessary to integrate, merge, align or integrate **o**ntologies. These processes collectively consist what is referred to as "interoperability" in the context of applied Ontology [283]. To this effect, there is much research on tools and methods of ontological interoperability [59, 63, 119, 129, 200, 299, 371, 372, 375]. As noted in chapter 3, there is diversity even attempting to achieve unity. For instance, several upper level **o**ntologies, such as SUMO and DOLCE are indeed designed with integration in mind [125]. The choice of a representation language for an **o**ntology is also itself seen as an important problem in the field [402]. Even if **o**ntologies are to be designed using a common language from the beginning, the question of what common language to pick is a debate in and of itself. Thus, it seems being pluralist is almost an inescapable reality for the working ontologist. And by the reduction thesis, being a logical pluralist is a necessity for ontological pluralism.

4.3.1 The added cost of monism

Despite the dramatic shift in perspective in the previous paragraph, some researchers in applied Ontology adhere to logical and ontological monism. The most prominent example of this was Douglas Lenat, who persistently pursued the objective of developing Cyc as the single, homogeneous all-encompassing **o**ntology. To a less extreme extent, Barry Smith is also an ontological and logical monist in that he touts BFO as the one upper-level **o**ntology, from which all other **o**ntologies should descend from, such as the Common Core or Industrial Ontologies Foundry **o**ntologies. [2]

Let us consider a hypothetical ontological monist as a thought experiment. By being an ontological monist, one is tasked to present the one true **o**ntology. Suppose one does find such **o**ntology via an ontological oracle. Now one faces the current situation — despite having the one true **o**ntology at hand, there are many other candidates in practice, being used in real-world applications. Accordingly, the ontological monist must choose to either disregard the other candidates, thereby adopting a form of ontological solipsism and accepting an ontological vacuum; or acknowledge the fact and devise a method to translate or subsume the existing **o**ntologies into the one true **o**ntology.

Suppose, on the other hand, one is an ontological pluralist. Rather than attempting to find the one true **o**ntology, one needs to establish or choose a framework for pluralism. Suppose the ontological pluralist chooses one via an ontological oracle. They are now tasked with developing tools and methods for integrating, merging, matching, translating, and subsuming **o**ntologies into other existing **o**ntologies. Note that the subsumption and translation of **o**ntologies is a requisite aspect of ontological pluralism.

Note that an ontological monist who does **not** embrace ontological solipsism will be required to develop a significant subset of the same methods and tools that an ontological pluralist must develop. In light of this pragmatic and instrumental observation, the debate between pluralism and monism debate, in the context of applied Ontology, may be framed candidly as follows: "ontological monism is simply pluralism with extra steps". Once again by the reduction thesis, "ontological" may be exchanged for "logical" throughout the entire argument sketch. Figure 4.1 depicts the argument as a flowchart.

4.3.2 The hidden price of pluralism

Since both monism and pluralism ultimately reach the same end status, provided one does not accept the ontological vacuum, what are the potential consequences of embracing of pluralism? The answer lies in the choice a pluralist framework. This is not a straightforward matter, as it becomes a question of choosing a variant of logical pluralism (by the reduction thesis). Such variant needs to be sufficiently flexible to align with the objectives of applied Ontology, yet not so flexible as to permit an "anything goes" scenario. The next chapter presents some variants of logical pluralism that have been proposed throughout history, with a focus on a select few promising candidates based on analysis of the existing literature. These promising candidates will then be explored in detail to generate sufficient machinery and tools for heterogeneous reasoning.

[2] Interestingly, Smith was the editor-in-chief for the journal *The Monist* from 1992 until 2016. Despite the name, the journal does not advocate for monism or any specific line or philosophical tradition as Smith himself has said [360, p. 158].

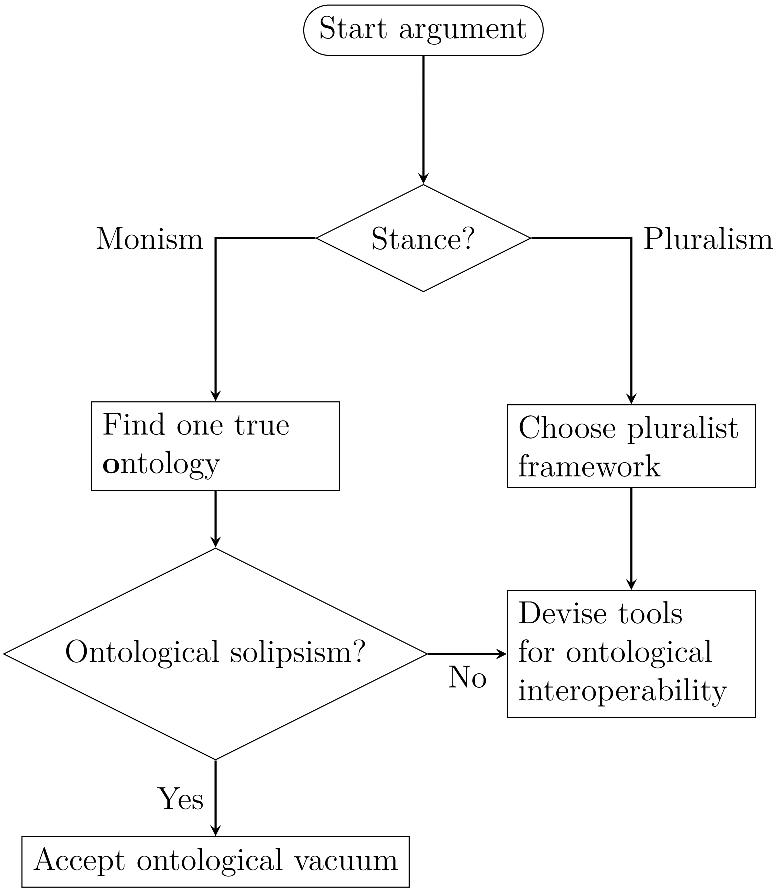


Figure 4.1: Sketch of argument for ontological pluralism.

This section closes the debate between pluralism and monism, as the debate itself is not within the scope of this work. For a more comprehensive and contextualized overview on the topic, the reader may refer to [93, 252, 331].

Chapter 5

From Ontology to Logic

This chapter initiates the interdisciplinary dialog proper between Logic and Ontology. It will present a historical overview of logic, with a particular focus on variants of logical pluralism. From the variants, two closely related instances are employed to spawn dual methodologies for logical and ontological integration. One of these methdologies is the existing Carnapian-Goguenist, while the other is the novel da Costian-Tarskianist.

Some previous knowledge on Logic and Category Theory is assumed throughout the chapter, at least from a conceptual point of view. On the matter of Logic, the unfamiliar reader may refer to [9, 173, 310, 321] for introductory accounts of varying depths covering propositional, first-order and some non-classical logics. The reader may refer to [238, 325] as introductions to Category Theory, while readers with a focus on logic may be interested in [140].

5.1 Historical Remarks

Logical pluralism has had a long and rich history [65, 93], it is therefore unsurprising that there exist several different interpretations of it. In broad strokes, logical pluralism accepts the existence of more than a single correct logic, usually taken to be "classical logic" as presented by Frege [124]. The idea of logical pluralism, alongside the debate between monism and pluralism, gained considerable momentum following the works of Carnap [71] on logic and language [65, p. 2].

A side remark: the term "classical" in "classical logic" is anachronistic, for it does not refer to the logic as practiced during what is known as classical antiquity in the discipline of History [151]. For instance, W. A. Carnielli, Pizzi, and Bueno-Soler [77, p. 25] highlight Aristotle wrote the first known treatment of modal logic in his *Organon*, despite living in classical antiquity. This does not mean he was a logical pluralist though — as Caret [65] pointed out, most logicians from Aristotle until Frege had the objective of presenting the *one true logic*, superior to all others. It is merely an aspect of Aristotle's one true logic that it had modal components in it, for the debate between pluralist and monist in logic was not considered.

5.1.1 Ways of logical pluralism

This section presents a non-exhaustive overview of the various forms of logical pluralism that acknowledge the monist versus pluralist debate. This account is by no means comprehensive, as the objective is to merely illustrate the considerable diversity that exists in the literature in preparation for the subsequent sections. The interested reader may refer to [66], [65] and [93] for additional references on the history of logical pluralism.

Vasiliev's "imaginary logic" tolerant of contradictions

Nicolai A. Vasiliev may be regarded as one of (if not the) first logical pluralist(s). V. A. Bazhanov [28] argues Vasiliev's works are at the turning point of the prehistory and history of paraconsistent logic, as he is the first to present a formal account of a logic devoid of the law of excluded middle. Vasiliev called his logic "imaginary logic" and rooted it in several areas of knowledge, from C. S. Peirce's logic to Charles Darwin's ideas on evolution. The "imaginary" aspect indeed comes from Lobachevsky's work on non-Euclidean geometry, which Vasiliev employs to describe the construction of his imaginary logic — via an imaginary geometry method [27]. In Vasiliev's logic, there are three kinds of partial propositions[1]: affirmative, negative and accidental. From the three kinds of propositions, Vasiliev presents a *new kind* of negation and thus conceives a logic free of the classical laws of the excluded middle and non-contradiction.

It is interesting to note that Vasiliev had views of the connections between Ontology and Logic. Namely, he posited that Logic as an area of study has an inherent ontological aspect which dictates what sort of logics one may stumble upon. As per V. A. Bazhanov [28], with his emphasis:

> Vasiliev persistently stressed the **primacy of an ontological aspect of logic**. By changing the ontology, combining the features of reality, we can get different imaginary logics, since the method of imaginary logic opens up the possibility of experimentation in logic, of giving up certain logical principles and seeing what comes of this rejection. This method resembles the "experimental methods of the natural sciences" (Vasiliev, 1912, p. 20). [28, p. 4]

In order to establish exactly what sorts of logics are acceptable or may be created via the imaginary method, Vasiliev relies on his concept of "metalogic": the laws enabling proposition and reasoning or the science of structures valid for every logical system. Among these laws, Vasiliev mentions the law of non-self-contradiction (an assertion cannot be true and false simultaneously) and *tertium non datur* (an assertion is either true or false). In other words, according to him, there are two logical levels — the metalogical level which is non-negotiable and the ontological level which can be varied and may, for instance, not include the classical law of non-contradiction (i.e. anything follows from an assertion and its negation). Vasiliev's imaginary logic leverages this two-level distinction to deny the law of non-contradiction at the ontological level and thus is "tolerant" of contradictions.

[1]Propositions of the form "Some S are (not) P".

Vasiliev is regarded as a pluralist *de facto* for not only acknowledging the plurality of multiple logical systems but also presenting a framework to characterize and generate new logical systems. As per V. A. Bazhanov's translation of Vasiliev's writings:

> "I am very well aware of the fact, - wrote Vasiliev in 1912, - that my idea of new logic contradicts the millenial conviction of mankind... I'm risk [*sic*] falling under the charge of logical heresy" (Vasiliev, 1912, p. 246). [27, p. 8]

Girard's unification of classical, intuitionistic and linear logics

In [136], J. Girard presents a view of logical pluralism motivated by the objective of unifying different logics into a common system, where logics are not disjoint and may interact with each other. Although it could initially be seen as a monist view, viz. an attempt to generate the one true logic, J. Girard is clear to state the goal of unification is to be able to view logics as fragments of a larger system and indeed establish them on equal reasoning footing — as he puts it, to allow a classical theorem to have an intuitionistic corollary and vice versa [136, p. 202].

To accomplish unification, J. Girard [136] introduces a sequent calculus named **LU**. **LU** embeds three different logics inside itself, namely classical, intuitionistic, and linear logic — the last of which being a sort of refinement of both classical and intuitionistic logics defined in [135]. In order to compartmentalize **LU** into the three fragments, J. Girard introduces the concept of polarized formulas: positive, neutral and negative formulas. The polarity of formulas adds a new semantic layer to **LU**, as each connective has a polarity table assigned to alongside its truth table. The polarity of a formula determines which sequent rules are applicable, an aspect of **LU** that may be understood intuitively as generating the three fragments. Given that the fragments are all within the same system, the fragments are able to interact with one another.

Laurent later expanded on the overall idea and mechanisms of embedding different logics as fragments of a larger system in his doctoral thesis [218]. His work gave rise to polarized proof theory and, by extension, several works using unification and polarization techniques [86, 219, 220, 229].

Prawitz's "logical ecumenisms"

Prawitz [306]'s view of logical pluralism is rooted in the assumption that the meaning of logical constants is circumstantial and depends on what one wants to say with one's logical sentences, a view he has consistently held throughout his works (see, for instance, [307]). In particular, Prawitz's concrete objective in [306] was to reconcile or harmonize the different meanings intended by the same logical symbols in intuitionistic and classical logic.

In order to achieve such reconciliation, Prawitz proposed a natural deduction system where both classical and intuitionistic logic co-exist by sharing certain logical symbols (the universal quantifier $\forall$, the conjunction $\wedge$, the negation $\neg$ and the absurd constant $\bot$), but have their own symbols for existential quantifiers $\exists_i, \exists_c$, disjunctions $\vee_i, \vee_c$ and implications $\rightarrow_i, \rightarrow_c$ (where the index c denotes the classical symbol and i denotes the intuitionistic symbol). Prawitz called these symbols "ecumenical" if they do not collapse into one another and allow intuitionistic and classical logicians to reason about both logics in the same mixed system:

> If they are sufficiently ecumenical and can use the other's vocabulary in their own speech, a classical logician and an intuitionist can both adopt the present mixed system, and the intuitionist must then agree that $A \vee_c \neg A$ is trivially provable for any sentence A, even when it contains intuitionistic constants, and the classical logician must admit that he has no ground for universally asserting $A \vee_i \neg A$, even when A contains only classical constants. That would require a general method for finding for any A a canonical proof of $A \vee_i \neg A$ whose immediate sub-proof must be either a proof of A or a proof of $\neg A$, and we do not know any such method. [306, p. 15]

Prawitz's pluralism via ecumenical systems has since been extended to other kinds of logics, such as modal logics in [247], and other mechanisms to combine logics [297]. The reader can refer to [298] for a recent survey on Prawitzian ecumenism and the systems it spawned.

Beall and Restall's case-based pluralism

J. Beall and Restall [30]'s logical pluralism rests on the tenet that logical consequence, as opposed to logical truth, is the main subject matter of Logic as discipline. That is, different logical systems have different notions of logical consequence and these notions are not competing, instead they are to be set on equal footing as far as logical citizenship. Furthermore, the validity of a logical argument depends on the underlying notion of consequence one is using. This is best encapsulated by the Generalized Tarski Thesis, as phrased by Russell [331]:

Thesis 5.1 (Generalized Tarski Thesis (GTT))**.** An argument is valid$_x$ if and only if in every case$_x$ in which the premises are true, so is the conclusion.

The precisification of what is meant by case$_x$ is, according to J. Beall and Restall [30], what characterizes a logic — viz. what are the models one is considering. Additionally, J. Beall and Restall note that pluralism arises when one assumes that the Generalized Tarski Thesis is true and that case$_x$ may be specified in at least two, equally acceptable, ways. This take on pluralism has since been termed case-based pluralism.

J. C. Beall and Restall's works are seminal, in that thy state a clear position on logical pluralism [29], and later expand the literature by further defending their position [29] and scrutinizing the meaning of "case", "valid", and "consequence" [31]. However, the proposal for case-based pluralism is predominantly assertive and not necessarily concerned with providing methods for assessing, combining or relating logics. This in contrast with the work by Prawitz [306] and J. Girard [136], both of which are concerned with concrete, specific logics. J. Beall and Restall's proposal, on the other hand, is broad and far-reaching.

As a matter of fact, due to its broad philosophical span and openness to interpretation (e.g. what exactly should constitute "every" or "case" in the Generalized Tarski Thesis), case-based pluralism has received several objections. It is not within this chapter's scope to increase the conceptual granularity and present further aspects of case-based pluralism or its objections, however the reader may refer to [331] for an overview. Case-based pluralism may be summarized as J. Beall and Restall [30] have done themselves:

> Logic is a matter of truth preservation in all cases. Different logics are given by different explications of these cases. This account of the nature of logical consequence sheds light on debates about different logics. Once this realisation is made apparent disagreements between some formal logics are shown to be just that: merely apparent. A number of different formal logics, in particular, classical logics, relevant logics and intuitionistic logics, have their place in formalising and regulating inference. Each is an elucidation of our pretheoretic, intuitive notion of logical consequence. [30, p. 17]

Varzi's logical relativity

Varzi [389]'s take on logical pluralism both extends and departs from J. Beall and Restall's case-based pluralism. Varzi argues that indeed logics are characterized by the class of all possible cases, or as he puts it more precisely, models to interpret a given language. However, Varzi's view is that there is no unique way of settling on how interpretation should work for a language and there is no unique way to define a logical language modelled after natural language.

In fact, Varzi is clear to state he is aligned with Tarski et al.'s liberalism and relativism when it comes to choosing logical constants:

> The relevant claim is that all (or any) terms of the language could in principle be regarded "as logical" — and I agree with that. [389, p. 5]

Varzi takes this position and attempts to generalize it as far as possible by presenting an abstract characterization of the semantic aspect of logics through Ajdukiewicz and Lambek's categorial grammars [3, 214]. Varzi's technical account will not be presented in its entirety, for it is quite involved. However, it is necessary to highlight that the objective with such technicality is to pave way for the possibility of a general relativistic semantic framework, and to strengthen Varzi's position on relativism:

> This concludes the technical point, which [...] should establish the claim that the distinction between logical and extra-logical terms is ultimately ungrounded, hence the claims leading to what I have called Tarskian Relativism. [389, p. 18-19]

Overall, Varzi's logical relativism allows for a plurality of logics to be considered correct as there are several, equally correct ways to choose the logical constants of a logic [378]. This focus on logical connectives has more recently been echoed in the work of Kissel [206], whose pluralist position is that not only the choice of logical constants, but the choice of their meaning gives rise to equally correct logics, if choosing their meaning is done under a particular framework.

Shapiro and Cook's logic-as-modelling pluralism

Cook and Shapiro depart from the view that Logic should be concerned with logical consequence and suggest that the point of a formal logic is to model a natural language [93, 350, 351]. The term "model" does not refer particularly to the semantic aspect of logics (as in model theory), but rather to a concept more closely aligned with that of models in Physics for explaining natural phenomena or linguistic models for explaining natural language structure (such as grammar, semantics, pragmatics). On the matter of models, Shapiro [351] writes:

> Of course, a model should also be "realistic" in that some of its features do correspond, more or less, to features of what it is a model of—logical aspects of natural languages in the present case. There should be a balance between simplicity and closeness of fit. With models generally, it is usually not a question of "getting it exactly right," even if the purposes and aspects being modeled are kept fixed. For a given purpose, there may be bad models—models that are clearly incorrect—and there may be good models, but it is unlikely that one can speak of the "correct model." There is almost always a gap between a model and what it is a model of. In most cases, one can make a model more "realistic" (i.e., more correct) at the cost of making it more cumbersome to work with and more difficult to study and use. [351, p. 48]

It may be noted that view of modelling and its ensuing matters of correctness could relate to Box's aphorism "all models are wrong", initially stated (albeit in different wording) in [50] in the context of statistical models. This aphorism encodes the contrast of epistemic truth versus epistemic usefulness, a topic found in the works of several writers prior, such as [316] and [393].

Since logics are to be seen as models, there may be several competing logics for modelling the same language. The question of the correctness of a logic could then become stratified. This means that a logic would be subject to degrees of correctness: comparable or non-comparable, absolute or relative to an objective. This could lead one to assume the position of logical nihilism, a view positing that there are no correct logics whatsoever. In order to avoid logical nihilism, Russell [331] notes Cook suggests two possible kinds of pluralism from the logic-as-modelling view: the goal-relative view, where correctness of a logic depends on an externally defined goal; and an extension of the goal-relative view, where two logics could be equally correct with respect to a goal.

Carnap's principle of tolerance

Rudolf Carnap is regarded as one of the forerunners on the matter of pluralism, whose work is considered by Caret [65, p. 2] as the first of three milestones of logical pluralism. Indeed, Carnap's so-called Principle of Tolerance is one of the first prototypical approaches to logical pluralism [324]. It is defined rather bluntly as follows [71] in *The Logical Syntax of Language*:

Definition 5.1.1 (Carnap's Principle of Tolerance)**.** It is not our business to set up prohibitions, but to arrive at conventions. [71, p. 52]

Carnap further elaborates his principle in the following passage which is often-quoted alongside the definition:

> In logic, there are no morals. Everyone is at liberty to build up his own logic, i.e., his own form of language, as he wishes. All that is required of him is that, if he wishes to discuss it, he must state his methods clearly, and give syntactical rules instead of philosophical arguments. [71, p. 52]

Carnap's principle is quite bold in that there is no mention of any particular "rules of thought" or meta-logic underlying how logics must be build. Taken at face value, it may be interpreted as a descriptive manifesto on formalizing thought for sharing purposes. Additionally, it may be noted that Carnap mentions *syntactical rules* in his Principle of Tolerance. Historically, Carnap's work pre-dates formal model theory. His aim was, in fact, to formalize *semantics* and scientific practice [68, 69, 223].

According to Caret [65, p. 3], Carnap's view is very much instrumental — accepting a given logic depends on how efficient it is as an instrument relative to a what Carnap calls a "linguistic framework". Carnap introduced the concept of linguistic framework in "Empiricism, Semantics and Ontology" to refer to a system of rules over terms and predicates devised to speak and reason about a particular subject or set of entities. For instance, a linguistic framework the terms and predicates used when talking about numbers or a physics theory. As noted by Price [309], Carnap's objective with linguistic frameworks is to separate internal from external questions: internal questions may be answered by reasoning within the framework itself, whereas external questions rest outside the reasoning capabilities of the framework.

For Carnap, the question of which logic is the one true logic, if in fact there is one, is external relates to the matter of electing a linguistic framework as the one true linguistic framework. This would require a linguistic framework to be capable of answering whether it is correct itself, a question which resides in the domain of external questions. Therefore, the matter of choosing a framework or a logic is a pragmatic matter, it depends on how well suited a given logic is based on the goals it is supposed to accomplish.

Arroyo and Silva [11] extract a different excerpt from Carnap's work to illustrate and elaborate on his Principle of Tolerance:

> [1] Let us grant to those who work in any special field of investigation the freedom to use any form of expression which seems useful to them; [2] the work in the field will sooner or later lead to the elimination of those forms which have no useful function. [67, p. 40]

Based on this excerpt, Arroyo and Silva [11] argue Carnap's view is pragmatic and also empirical, since it is actually motivated by the quest of dealing with different scientific frameworks. That is, the choice between two logics (or, two scientific frameworks) depends on conventions and observations of what *works* in practice. Arroyo and Silva [11] agree with Caret [65] in that Carnap's view (framed as linguistic realism in [11]) is in line with the "logic as calculus" view:

> What Carnap's linguistic realism cannot do is to specify a fact of the matter about what exists in reality, so the realism is confined with existence questions that don't go beyond any given linguistic framework. [11, p. 15]

W. V. Quine [313] and Popper [304] have widely criticized Carnap's views[2], which in turn led a decline in interest pertaining to Carnap's works. Nevertheless, his views continue to exert a significant influence on the literature. Section 5.2.1 details how Carnap's pluralist perspective gave rise to the implementation of an approach of ontological heterogeneity.

da Costa's principle of non triviality

Newton Carneiro Affonso da Costa is widely acknowledged as the founder of the so-called Brazilian school of paraconsistency for his pioneering work on inconsistent (but non-trivial) formal systems and for presenting formal calculi for reasoning over such inconsistent systems [99, 100]. da Costa is also renowned for developing the concept of *quasi-truth*, a generalization of Alfred Tarki's definition of truth. This concept attempts to capture the meaning behind theories of certain pragmatist thinkers such as Charles Sanders Peirce and William James [256]. It is noteworthy that the underlying logic of quasi-truth has been demonstrated to be paraconsistent, as evidenced by D'Ottaviano and Hifume [106], highlighting a connection between the two concepts.

Perhaps less well-known is da Costa's perspective on logical pluralism proper. As D'Ottaviano and Gomes [105] note, da Costa has consistently sought to place inconsistent and consistent systems on equal footing, even prior to presenting paraconsistent logics in his thesis. da Costa was in fact inspired by Carnap's Principle of Tolerance, and coined his own Principle of Tolerance *in Mathematics* in [98]. The following is a translation of an excerpt where da Costa first enunciates Principle of Tolerance in Mathematics. Emphasis and casing come from [105].

> Summing up all forethought, we propose a *Principle of Tolerance in Mathematics*, akin to Carnap's in Syntax, enunciated as follows: From the syntactical-semantical point of view, every mathematical theory is admissible so long it is not trivial. In broad strokes, there *exists* in Mathematics all which *is not trivial.*

da Costa's principle is very keen on wording to characterize *non-trivial* theories as admissible. Drawing from this fact, W. A. Carnielli and Marcos [75] discuss the relationship between the concepts of contradictoriness, inconsistency, explosiveness and triviality — triviality entails contradictoriness and inconsistency, however the opposite does not necessarily hold. Indeed, as W. A. Carnielli and Marcos [75, p. 19] note, trivial logics are not particularly useful for the purpose of modeling reasoning, since any statement holds in such logics. Consequently, they propose renaming da Costa's principle to the Principle of Non-Triviality. In any case, this pragmatic observation is in accordance with da Costa's actual view, as evidenced by the following translated excerpt from his thesis [100]:

> [...] for several reasons, such as, for example, comparative analysis of consistent systems and correct valuation, from the metamathematical point of view, of the several principles at stake, it is then convenient to study inconsistent

[2]It may be noted, however, that Popper misrepresents Carnap's position. Michalos [255] writes a brief review on the topic.

> systems in a direct manner. But, for that, it is a requirement to structure new kinds of elementary logic, with the help of which it is possible to manipulate such systems.

In a manner of speaking, da Costa's Principle of Tolerance, or Principle of Non-Triviality as it shall be referred to heretoafter, represents a kind of duality to Carnap's Principle of Tolerance. The former is concerned with *syntax*, since triviality is a purely syntactical concept, whereas the latter is concerned with semantics. The chapter 6 expands on this duality by discussing W. A. Carnielli and Marcos's formalization of da Costa's Principle of Non-Triviality via consequence relations.

5.2 Towards heterogeneity

It should be noted that a philosophical view of logical pluralism provides a means to accept and acknowledge a diversity of logics. However, in general, such a view does not provide mechanisms for logics to interact. For this reason, in the context ontological interoperability, it is necessary to develop frameworks for implementing not only logical pluralism but also logical and, by the reduction thesis from chapter 4, ontological *heterogeneity*. Logical heterogeneity differs itself from logical pluralism in that it extends beyond mere acceptance, toleration, acknowledgement or support of different kinds of logics. In order to implement or develop heterogeneity, it is necessary to present methods for integrating, refining, splicing, slicing, joining, and in general, manipulating different logics.

The preceding section presents several prespectives on logical pluralism exhibiting a spectrum of formalization or, as it may be termed, mechanization of pluralism. For example, Girard's view of logical pluralism led to polarized proof theory, which provides mechanisms for unifying logics. Conversely, it could be argued that the case-based approach proposed by Beall and Restall, and logic-as-modelling perspective by Shapiro and Cook's, do not intend to provide such a set of mechanisms. In light of these considerations, the suitability of adopting a pluralist view, say P_x, as the foundation of a heterogeneous framework is contingent upon how opinionated P_x is in terms of formalizing or representing logics, as well as the ease with which P_x may be extended into a framework.

This section presents Lücke's heterogeneous framework which extends Carnap's pluralist view through Goguen's institution theory [235], thus being termed "Carnapian-Goguenism". Carnapian-Goguenism illustrates how a particular choice of pluralist view may lead to a framework, however it should not be taken as the only possible framework. This section serves, therefore, as a preamble for the novel framework, termed da Costian-Tarskianism, presented in chapter 6. Da Costian-Tarskianism in fact heavily draws from Carnapian-Goguenism, hence why such attention is taken to reproduce this framework in the following subsection.

It may be noted that each heterogeneous framework may offer different definitions of what constitutes a logic, a concept that has been largely overlooked until now, and what is an **o**ntology. This means that heterogeneous frameworks may formally abandon Guarino's definition from chapter 2. However, it is desirable, in general, to retain some of the key aspects of Guarino's definition.

5.2.1 The Carnapian approach

The Carnapian approach towards ontological heterogeneity stems from the work developed by members of the Common Framework Initiative (CoFI) over the course of more than a decade (see [38, 333] for some history on CoFI), culminating in the creation of the Distributed Ontology, Model and Specification Language (DOL) [263]. It is noteworthy that none of the CoFI members had initially been concerned with applied Ontology; rather, their focus was on algebraic specification matters. It was only much later, in [235] proper, that ontological specification received its due attention. This further highlights the pervasiveness of the reduction thesis, as discussed in chapter 4.

What is a logic?

Since the purpose of a heterogeneous logical framework is to provide tools to manipulate logics, it is first necessary to define what is, in fact, a logic. To answer this, Mossakowski et al. [266] re-interpret *institution theory* to provide a category-theoretical definition of what is a logic. Their approach sits within the domain of Universal Logic, an analogy to Universal Algebra (i.e. the mathematical study of properties common to all algebraic structures) popularized by Béziau [37].

J. A. Goguen and Burstall [139] originally utilized institutions in the context of formal software specification, as a very abstract way of specifying computer programs. Moreover, for practical reasons, they did not originally use category-theoretical constructs to define what is an institution is. Despite this, this section will present a category-theoretical definition following [235].

The notation in this section is fairly standard in the category theory literature. Given a category C, $|C|$ denotes its objects and C^{op} denotes its opposite category. Given two morphisms f and g from C, $g \circ f$ denotes composition. The category **Set** denotes the category whose objects are small sets and morphisms are functions, and $\mathbb{CAT}$ is the quasi-category of all categories.

Definition 5.2.1 (Institution, as per [266])**.** An *institution* is a 4-tuple

$$I = \langle \mathbf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \models \rangle$$

where

- **Sign** is a category of *signatures* and *signature morphisms*[3]
- **Sen** : **Sign** $\rightarrow$ **Set** is a functor assigning for each signature Σ, the set of its sentences $\mathbf{Sen}(\Sigma)$ and for each signature morphism $\sigma : \Sigma \rightarrow \Sigma'$, the sentence translation map $\mathbf{Sen}(\sigma) : \mathbf{Sen}(\Sigma) \rightarrow \mathbf{Sen}(\Sigma')$.
- **Mod** : $\mathbf{Sign}^{op} \rightarrow \mathbb{CAT}$ is a functor assigning each signature Σ to a category of models $\mathbf{Mod}(\Sigma)$ over Σ and each signature morphism $\sigma : \Sigma \rightarrow \Sigma'$ to a reduct

[3] Lücke [235] deliberately do not specify what are the actual objects and morphisms of **Sign**. The motivation for this is that signatures of an institution are chosen based on the logic one wants to represent, thus different institutions (representing different logics) have different categories of signatures. Moreover, it is not necessary to specify a highly abstract category of signatures (as presented, for instance, by Jacobs [195])for any and every institution.

functor $\mathbf{Mod}(\sigma) : \mathbf{Mod}(\Sigma') \to \mathbf{Mod}(\Sigma)$. $\mathbf{Mod}(\sigma)(M')$ may be written as $M' \restriction_\sigma$ and be called the *σ-reduct* of M', while M' is called the *σ-expansion* of $M' \restriction_\sigma$.

- $\models_\Sigma$ is a satisfaction relation, $\models_\Sigma \subseteq |\mathbf{Mod}(\Sigma)| \times \mathbf{Sen}(\Sigma)$ for each $\Sigma \in |\mathbf{Sign}|$. Such that for each $\sigma : \Sigma \to \Sigma'$ in **Sign**, the following satisfaction condition holds:

$$M' \models_{\Sigma'} \mathbf{Sen}(\sigma)(\phi) \text{ iff } \mathbf{Mod}(\sigma)(M') \models_\Sigma \phi$$

for each $M' \in |\mathbf{Mod}(\Sigma')|$ and $\phi \in \mathbf{Sen}(\Sigma)$.

It is necessary to make a few remarks about this definition. An institution captures the **semantic aspect** of a logic and, as Mossakowski et al. [266, p. 4] note, is that all components of an institution are parameterized by signature. The signature of an institution does **not** contain logical symbols (for example, $\wedge$, $\vee$, etc. from propositional calculus). It is the role of the **Sen** functor to implicitly encode the logical symbols. Note that an institution is devoid of any syntax-related constructs, such as a consequence operator or an entailment system. For this reason, a single institution corresponds to an entire *class* of logics — when one attaches a syntactical construct (i.e. a consequence operator or an entailment system) to an institution, it then becomes a signature-agnostic logic proper. See the definition of logic presented by Meseguer [254] for a concrete formal account of this idea.

In addition, the informal reading of the satisfaction condition can be stated as follows: if a model satisfies a certain sentence under a particular satisfaction relation, then the translated model also satisfies the translated sentence, and vice versa. In other words, truth is invariant under change of notation and context. The keen reader may note that the **Mod** functor could also be defined in terms of the **Sign** category instead of its opposite, i.e. one could define $\mathbf{Mod}' : \mathbf{Sign} \to \mathbb{CAT}$. In this case, the satisfaction condition would be stated as follows: for each $\sigma : \Sigma \to \Sigma'$, $\mathbf{Mod}'(\sigma)(M) \models_{\Sigma'} \mathbf{Sen}(\sigma)(\phi)$ if and only if $M \models_\Sigma \phi$. The rationale for employing the opposite category will become evident in the subsequent discussion about morphisms between institutions.

Institutions are able to represent a wide array of different logics. The following are some examples of logics formalized as institutions, the majority of which comes from [235].

Example 5.2.1. Propositional logic can be formalized via the following institution $\mathbf{Prop} = \langle \mathbf{SetSign}, \mathbf{PropSen}, \mathbf{PropMod}, \models \rangle$. Here is how each component is defined:

- The category **SetSign** has as its objects sets Σ of propositional symbols and as morphisms functions $\sigma : \Sigma_1 \to \Sigma_2$. In this case, $\top$ and $\bot$ are taken to be be propositional symbols as well, represented by *true* and *false* respectively.

- $\mathbf{PropSen} : \mathbf{SetSign} \to \mathbf{Set}$ maps each signature Σ to the usual set of propositional sentences[4] $\mathbf{Sen}(\Sigma)$ and maps each signature morphism $\sigma : \Sigma_1 \to \Sigma_2$ to a morphism $L(\Sigma_1) \to \mathbf{Sen}(\Sigma_2)$ replacing each propositional symbol of Σ_1 by the corresponding symbol of Σ_2 according to σ.

- $\mathbf{PropMod} : \mathbf{SetSign}^{op} \to \mathbb{CAT}$ maps each signature Σ to a parameterized category $\mathbf{PropMod}_\Sigma$ whose objects are functions $\Sigma \to \{true, false\}$, also called Σ-models.

[4] This can be obtained, for instance, via the closure of all usual propositional and logical symbols.

Additionally, **PropMod** maps each signature morphism $\sigma : \Sigma_1 \rightarrow \Sigma_2$ to a functor $\mathbf{PropMod}_{\Sigma_1} \rightarrow \mathbf{PropMod}_{\Sigma_2}$ which reduces a Σ_2-model M_2 to a Σ_1-model M_1 by composing $M_2 \circ \sigma$.

The satisfaction relation $\models$ is defined according to standard truth-table semantics, meaning truth is evaluated in a bottom-up manner by evaluating propositional symbols.

Example 5.2.2. As institutions are parameterized by signatures, it is challenging to provide a concrete example. One can instead consider a toy (instance of an) institution, for illustrative purposes. Let us fix a signature category $\mathbf{PropSign_3}$ containing only three objects, A, $true$ and $false$ and no morphisms[5]. The corresponding concrete institution $\mathbf{Prop}_3$ is defined as the tuple $\langle \mathbf{PropSign}_3, \mathbf{PropSen}_3, \mathbf{PropMod}_3, \models_3 \rangle$. The functor $\mathbf{PropSen}_3$ generates, so-to-speak, all sentences using the propositional symbols from $\mathbf{PropSign_3}$ and logical symbols, i.e. sentences such as $A \wedge A$, $A \wedge \neg A \vee true$, $false \vee A \rightarrow true, \ldots$ $\mathbf{PropMod}_3$ generates two models for the signature $\mathbf{PropSign}_3$, namely two functions f_1, f_2 such that $f_1(A) = true$ and $f_2(A) = false$. The role of the actual **Prop** institution, from the previous example, is to generalize over this toy concrete institution.

Example 5.2.3. Untyped first-order logic with equality can also be represented via an institution $\mathbf{FOL}^=$. The signatures are first-order signatures, tuples consisting of a set of function symbols with arities and a set of predicate symbols with arities, and signature morphisms map symbols while preserving arities. Models are usual first-order structures and sentences are usual first-order formulas. Sentence translation means replacing translated symbols, and model reduction (or translation) means restructuring a model to fit the signature morphism. Satisfaction is the usual satisfaction of a first-order sentence given a first-order structure. One can "extend" the institution $\mathbf{FOL}^=$ to obtain another institution $\mathbf{FOL}^{\mathrm{ms}=}$ representing many-sorted first-order logic (see [139] for more details on these institutions). The top-level **o**ntology DOLCE, from chapter 3, has in fact been formalized using first-order logic.

Example 5.2.4. It is also possible to represent modal logics using institutions. Consider the institution **K** representing the modal logic (i.e. propositional logic equipped with a single modality operator and a single extra axiom). The signature of **K** is **SetSign**, the sentence functor is similarly defined as **PropSen** adding the unary $\Box$ operator. Models of **K** are standard Kripke models (corresponding to sets with accessibility relations) and satisfaction is standard modal satisfaction. Other modal logics can also be represented similarly, such as **S4** and **S5**.

The modal first-order logic **QS5**, also used to formalize DOLCE, can be formalized using institutions. It can be regarded as a combination between **S5** and $\mathbf{FOL}^=$. The signature of **QS5** is the same as that of $\mathbf{FOL}^=$, the sentences are also similar to those of $\mathbf{FOL}^=$ including boolean quantifiers, identity and adding the $\Box$ operator while removing constants and function symbols. Predicate symbols in **QS5** can be marked as either flexible or rigid with respect to the domain. Models are constant-domain first-order Kripke structures, with usual first-order modal satisfaction. Because modal logics can actually be seen as fragments of first-order logic [387], combining modality and quantification entails many syntactical and semantic complications as Lücke [235, p. 23] note.

Example 5.2.5. Description logics can be formalized as institutions. We will look at the simplest example, $\mathcal{ALC}$ and defer a deeper discussion for chapter 7. This example has

[5]This category can also be called a discrete category.

been extracted nearly verbatim from [235, p. 273]. Signatures of the description logic $\mathcal{ALC}$ consist of a set B of atomic concepts and a set R of roles, while signature morphisms provide respective mappings. Models are single-sorted first-order structures that interpret concepts as unary and roles as binary predicates. Sentences are subsumption relations $C_1 \sqsubseteq C_2$ between concepts, where concepts follow the grammar

$$C ::= B|\top|\bot|C1 \sqcup C2|C1 \sqcap C2|\neg C|\forall\ R.C|\exists\ R.C$$

Sentence translation and reduct is defined similarly as in $\mathbf{FOL}^=$. Satisfaction is the standard satisfaction of description logics. The reader can refer to [234] for a deeper institutional-theoretic account of description logics.

As noted by Mossakowski et al. [266, p. 5] Many other families of logics can be represented using institutions, such as description, higher-order, paraconsistent, substructural, polymorphic, coalgebraic, temporal, object-oriented logics, Common Logic and relational schemes. Chapter 7 will revisit some of these logics.

Two institutions can isomorphically represent the same logic[6], a fact which may be regarded as an issue from the philosophical point of view. In order to address this issue, Mossakowski et al. [266] present the concepts of institution morphism and comorphism with the intention of providing a more definitive answer to the question of what is the *identity* of a logic. In a subsequent paper, the same authors [264] address the question of what is a logic translation by using institution comorphisms (originally from [137]). A logic translation in the institutional-theoretic framework is defined as an institution comorphism itself as follows:

Definition 5.2.2 (Institution Comorphism)**.** Given two institutions I and J with $I = \langle \mathbf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \models \rangle$ and $J = \langle \mathbf{Sign}', \mathbf{Sen}', \mathbf{Mod}', \models' \rangle$, an *institution comorphism* from I to J consists of a functor $\Phi : \mathbf{Sign} \to \mathbf{Sign}'$ and natural transformations $\beta : \mathbf{Mod}' \circ \Phi \Longrightarrow \mathbf{Mod}$ and $\alpha : \mathbf{Sen} \Longrightarrow \mathbf{Sen}' \circ \Phi$ such that the following *satisfaction condition* holds

$$M' \models^J_{\Phi(\Sigma)} \alpha_\Sigma(\phi) \Longleftrightarrow \beta_\Sigma(M') \models^I_\Sigma \phi$$

Let us dive into each component:

- $\Phi(\Sigma)$ is the translation of signature $\Sigma \in |\mathbf{Sign}|$ from institution I into a signature $\Sigma' \in |\mathbf{Sign}'|$ from institution J.
- $\alpha_\Sigma(\phi)$ is the translation of a Σ-sentence $\phi \in |\mathbf{Sen}(\Sigma)|$ from I into a $\Phi(\Sigma)$-sentence $\phi' \in |\mathbf{Sen}'(\Phi(\Sigma))|$ from J
- $\beta_\Sigma(M')$ is the translation (i.e. reduction) of a $\Phi(\Sigma)$-model $M' \in |\mathbf{Mod}'(\Phi(\Sigma))|$ from J to a Σ-model $M \in |\mathbf{Mod}(\Sigma)|$ from I.

Given that two institutions I and J formalize two classes of logics, the intuitive interpretation of an institution comorphism is that it maps one institution to another while preserving satisfaction in each institution, i.e. the satisfaction condition. It should be noted that this notion of translation is based solely on preservation of truth. A subinstitution, originally defined by [254], is a particular kind of institution comorphism defined as follows:

[6] Technically, the same class of logics.

Definition 5.2.3 (Subinstitution)**.** A *subinstitution* is an institution comorphism such that Φ is an embedding of categories, α_Σ is injective and β_Σ is an isomorphism for every Σ.

Subinstitutions capture the intuitive notion of having one logic inside another. For instance, propositional logic is usually seen as being "part of" or "embedded inside" untyped first-order logic. This is precisely represented by a subinstitution between **Prop** and $\mathbf{FOL}^=$.

What is an ontology?

Following up on [264, 266], Lücke [235] put forward a comprehensive framework for logical and ontological integration. Their view on logical and ontological integration is termed *Carnapian-Goguenism*, as an homage to both Rudolf Carnap and Joseph Goguen, since it is inspired by Carnap's Principle of Tolerance and Goguen's (and Rod Burstall's) work on institutions and "Semantics First!" motto[7].

Lücke [235] lay out their framework by redefining what is an **o**ntology and providing mechanisms for **o**ntologies to interact. The initial step towards their endeavor is presenting the concept of a development graph.

Definition 5.2.4 (Development Graph)**.** A *development graph* over an institution $I = \langle \mathbf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \models \rangle$ is a vertex and edge-labeled directed, acyclic graph (DAG) $\mathcal{DG} = \langle \mathcal{N}, \mathcal{L} \rangle$, where:

- $\mathcal{N}$ is a set of of nodes, where each node $N \in \mathcal{N}$ is labeled with a pair $\langle \Sigma^N, \Psi^N \rangle$ such that Σ^N is a signature and $\Psi^N \subseteq \mathbf{Sen}(\Sigma^N)$ is the set of *local axioms* of N.
- $\mathcal{L}$ is a sorted set of directed links:
  - *definition links* $K \xrightarrow{\sigma} N$ annotated with a signature morphism $\sigma : K \to N$;
  - *hiding definition links* $K \underset{h}{\xrightarrow{\sigma}} N$ annotated with a signature morphism $\sigma : \Sigma^N \to \Sigma^K$, going against the direction of the definition link;
  - *theorem links* $K \overset{\sigma}{\dashrightarrow} N$ for each $K, N \in \mathcal{N}$ such that the following property holds: for all $M \in \mathbf{Mod}(N)$, $M \restriction_\sigma \in \mathbf{Mod}(K)$.

Definition links state the theories of nodes that are linked, while theorem links postulate relations between different nodes. A link may be annotated to express it possesses a particular property.

Definition 5.2.5 (Link annotations)**.** A global definition (or theorem) link $K \underset{p}{\overset{\sigma}{\dashrightarrow}} N$ may be annotated with $p \in \{cons, mono\}$ to express it is:

- *cons*: a conservative extension link, meaning every K-model has a σ-expansion to an N-model

[7]This motto appears in a very direct manner in a never-published textbook by Goguen on theorem proving and algebra [138, p. 25-26].

- *mono*: a conservative extension link such that any isomorphism $h : A \restriction_\sigma \rightarrow B \restriction_\sigma B$ between reducts of N-models has a σ-expansion to an isomorphism $A \rightarrow B$ between K-models.

A development graph DG is necessarily defined over an institution I, however one can assume that I is *any* institution if it is not explicitly stated. Development graphs intuitively capture the notion of importing an **o**ntology into another whilst translating symbols. To be a bit more precise, suppose, for the time being, that an **o**ntology is a theory (i.e. a subset of $\mathbf{Sen}(\sigma)$) over an institution. Classes of models and theories are defined over a node of a development graph as follows:

Definition 5.2.6 (Class of models over a development graph)**.** Given a development graph $\mathcal{DG} = \langle \mathcal{N}, \mathcal{L} \rangle$ and a node $N \in \mathcal{N}$, its associated class of models $\mathbf{Mod}_{\mathcal{DG}}(N)$ (or N-models) is inductively defined as the Σ^N-models M for which:

- M satisfies the local axioms Ψ^N, i.e. $M \models \psi$ for each $\psi \in \Psi^N$
- for each $K \xrightarrow{\sigma} N \in \mathcal{L}$, $M \restriction_\sigma$ is a K-model
- for each $K \underset{h}{\xrightarrow{\sigma}} N \in \mathcal{L}$, M has a σ-expansion M' (i.e. $M' \restriction_\sigma = M$) that is a K-model.

Definition 5.2.7 (Theory over a development graph)**.** Given a development graph $\mathcal{DG} = \langle \mathcal{N}, \mathcal{L} \rangle$ and a node $N \in \mathcal{N}$, its associated theory $\mathbf{Th}_{\mathcal{DG}}(N)$ is inductively defined to consist of

- all the local axioms Ψ^N
- for each $K \xrightarrow{\sigma} N \in \mathcal{L}$, all of $\mathbf{Th}_{\mathcal{DG}}(K)$ translated by σ.

The class of $\mathbf{Mod}_{\mathcal{DG}}(N)$ of a node N corresponds to the models which satisfy the axioms Ψ^N under the satisfaction relation $\vdash_{\Sigma^N}$ and, additionally, the models which satisfy, under appropriate translations, all the theories of the nodes linked to N via definition and hiding links. Similarly, the theory $\mathbf{Th}_{\mathcal{DG}}(N)$ corresponds of a node N corresponds to the axioms Ψ^N plus all the axioms of the nodes linked to N via theorem links.

The notion of importing an **o**ntology into another can be made clearer now. If there is a definition link between two nodes K and N in a development graph, the theory of N must include the translated theory of K. If one interprets the theory of a node as being an **o**ntology, definition links can be understood as "injecting" an **o**ntology into another. It may be noted that this interpretation assumes both **o**ntologies have the same underlying institution.

The fact that development graphs are defined over a single institution means they are unable to represent heterogeneous theories. This is because an institution is have taken to be equivalent, in terms of semantics, to a single logic. In order to overcome this apparent limitation, Lücke [235, p. 285] introduce Grothendieck institutions, as initially developed by Diaconescu [113]. Grothendieck institutions are an instance of a Grothendieck construction, a category-theoretical apparatus to encode several categories inside a single category.[8]

[8]Spivak [369, p. 367] presents the following slogan for Grothendieck constructions: "The Grothendieck construction takes structured, tabulated data and flattens it by throwing it all into one big space. The projection functor is then tasked with remembering which box each datum originally came from."

Definition 5.2.8 (Grothendieck institution)**.** Fix an arbitrary graph of institutions and institutions comorphisms (called a *logic graph*). A *Grothendieck institution* is defined by its components as follows. A signature in the Grothendieck institution over this graph consists of a pair $\langle L, \Sigma \rangle$ where L is a logic formalized as an institution and Σ is a signature in the logic L. A Grothendieck signature morphism $\langle \rho, \sigma \rangle : \langle L_1, \Sigma_1 \rangle \to \langle L_2, \Sigma_2 \rangle$ consists of a logic translation formalized as an institution comorphism $\rho = (\Phi, \alpha, \beta) : L_1 \to L_2$ plus an L_2-signature morphism $\sigma : \Phi(\Sigma_1) \to \Sigma_2$. Sentences, models and satisfaction in the Grothendieck institution are defined in a componentwise manner.

Finally, **o**ntologies and *hyperontologies* are defined as follows:

Definition 5.2.9 (**O**ntologies and hyperontologies)**.** An *abstract structured heterogeneous ontology* with respect to some logic graph, viz. an **o**ntology, is a node O in a development graph $\mathcal{DG}$ whose underlying institution is a Grothendieck institution built over the same logic graph. A *hyperontology* is the entire development graph.

This new definition states an **o**ntology cannot exist in isolated form — that is, it exists as part of a development graph. Even if the development graph were to consist of a single node, it could be subjected to graph-theoretical operations such as adding more nodes and edges. This is in contrast to definition 2.2.12, which does not presuppose the existence of any underlying structure upon which **o**ntologies rest. Additionally, this breaks the so-called **o**ntological vacuum discussed in chapter 2, enabling the development of methods to relate and combine **o**ntologies.

How to relate ontologies?

Empowered by definition 5.2.9, Kutz and Mossakowski [212, p. 288-289] present three different possibilities of relating **o**ntologies:

- *Refinement*: an **o**ntology can be refined into another by specifying a mapping which translates the former into the latter.
- *Integration*: two **o**ntologies can be mapped into a third *existing* reference ontology.
- *Connection*: two **o**ntologies can be related via some additional interface **o**ntology, usually specified manually, which is used to *generate* an overall third **o**ntology.

Integration and connections are symmetric combination techniques in that the order in which the **o**ntologies participate in the combination is inconsequential. In contrast, refinements are asymmetric, for the mapping may not necessarily be reversible. A refinement operation captures the idea of mapping a "coarser" (or less expressive) **o**ntology into a "finer" (or more expressive) **o**ntology.

Refinement. Lücke [235, p. 289] define homogeneous and heterogeneous refinements, making a distinction between both. Homogeneous refinements follow the usual definition from specification theory, as presented in [16], while heterogeneous refinements are defined in a manner that is tailored to suit the Carnapian-Goguenist approach.

Definition 5.2.10 (Standard or homogeneous refinements)**.** Given two **o**ntologies O_1 and O_2 in the same logic (or institution), O_2 is called a *standard refinement* of O_1 if there is a theorem link $O_1 \overset{\sigma}{\dashrightarrow} O_2$ that follows from the underlying development graph.

Definition 5.2.11 (Heterogeneous refinement)**.** Given two **o**ntologies O_1 and O_2, O_2 is called a (*heterogeneous*) refinement of O_1 if:

- there is an **o**ntology O_2' (called a *monomorphic extension*) such that there exists a conservative monomorphic definition link $O_2 \xrightarrow[mono]{\sigma} O_2'$ and
- there is a theorem link $O_1 \overset{\sigma}{\dashrightarrow} O_2$

that follows from the underlying development graph. If the theorem link is conservative, the refinement is called *conservative* as well. This can be visualized via the following diagram:

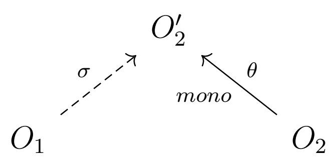


Figure 5.1: Heterogeneous refinement from O_1 to O_2.

The intuitive reading of heterogeneous refinements is they constitute means for refining **o**ntologies with "non-comparable" logics. For instance, if O_1 has sorts whereas O_2 does not, a heterogeneous refinement first finds a common super-logic O_2' whose theory is a super-set of that of O_1. Refinements have the following properties:

Proposition 5.2.1. *For a heterogeneous refinement*

$$O_1 \overset{\sigma}{\dashrightarrow} O_2' \xleftarrow[mono]{\theta} O_2$$

1. *any O_2-model can be translated to an O_1-model;*
2. *logical consequence is preserved along refinement, i.e. if $O_1 \models \phi$ then $O_2 \models \theta^{-1}(\sigma(\phi))$;*
3. *for conservative refinements, any O_1-model can be translated to an O_2-model;*
4. *the target of a conservative refinement has at least as many non-isomorphic models as the source.*

Proof. Due to Lücke [235, p. 290-291]. □

A sub-**o**ntology is also defined in terms of a refinement and depends on the concept of subinstitution.

Definition 5.2.12 (Sub-**o**ntology)**.** An **o**ntology O_1 is a (*heterogeneous*) sub-**o**ntology of O_2 if O_2 is a (heterogeneous) refinement of O_1 such that:

- the monomorphic extension is trivial (i.e. the identity),
- the signature morphism part of the theorem link is a monomorphism,
- the institution comorphism part of the theorem link is a subinstitution.

Refinements can compose under certain conditions, namely when the underlying Grothendieck institution of the development graph is *semi-exact*. This is defined as follows:

Definition 5.2.13 (Semi-exact institutions)**.** An institution is *semi-exact* if **Sign** has pushouts and, moreover, the model functor takes any pushout

$$\begin{array}{ccc} \Sigma & \longrightarrow & \Sigma_1 \\ \downarrow & \lrcorner & \downarrow \\ \Sigma_2 & \longrightarrow & \Sigma_R \end{array}$$

in **Sign** to a pullback

$$\begin{array}{ccc} \mathbf{Mod}(\Sigma) & \longleftarrow & \mathbf{Mod}(\Sigma_1) \\ \uparrow & \ulcorner & \uparrow \\ \mathbf{Mod}(\Sigma_2) & \longleftarrow & \mathbf{Mod}(\Sigma_R) \end{array}$$

of categories. In other words, for any pair $(M_1, M_2) \in \mathbf{Mod}(\Sigma_1) \times \mathbf{Mod}(\Sigma_2)$ that is *compatible*, meaning M_1 and M_2 reduce to the same Σ-model, can be *amalgamated* to a unique Σ_R-model M. Similarly, any pair of model morphisms that are compatible in the same sense can be amalgamated to a model morphism.

Semi-exact institutions refinements have the following property [235, p. 293]:

Proposition 5.2.2. *In semi-exact institutions, heterogeneous refinements compose.*

Proof. Composition is as follows, where the square is a pushout:

$$\begin{array}{ccccc} O_1 & \xrightarrow[mono]{\sigma_1} & O_2' & \dashrightarrow & O' & & \\ & & \uparrow \theta_1 & & \uparrow mono & & \\ & & O_2 & \overset{\sigma_2}{\dashrightarrow} & O_3' & \xleftarrow[mono]{\theta_2} & O_3 \end{array}$$

By semi-exactness, monomorphicity lifts along pushouts. □

Diaconescu [114] and Mossakowski [260] present the conditions under which an institution is semi-exact. Several institutions such as **Prop**, **FOL**, **QS5**, **K** are semi-exact.

Lücke [235] also define what it means for two **o**ntologies to be equivalent in terms of refinements. Intuitively, two **o**ntologies are equivalent if one can be directly or indirectly refined into the other, i.e. if one can be extended any number of times until it is refined into the other. If no extension steps are required, the **o**ntologies are called weakly equivalent. This is formalized via the two definitions that follow.

Definition 5.2.14 (Weak equivalence)**.** Two **o**ntologies O_1 and O_2 are called *weakly equivalent* if they can be conservatively refined into each other.

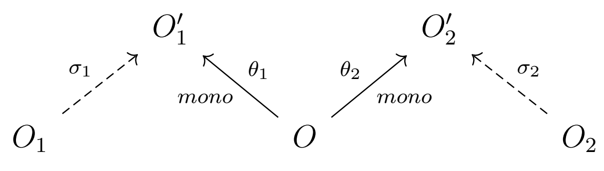
Definition 5.2.15 (Pre-equivalence and equivalence)**.** Two **o**ntologies O_1 and O_2 are called *pre-equivalent*, written $O_1 \approx O_2$, if there is a common monomorphic extension O of O_1 and O_2. *Equivalence* of **o**ntologies is defined to be the transitive closure of pre-equivalence.

Lücke [235] present some properties of (weak or pre-)equivalence in section 3.1.2.

Integration. The second possibility of **o**ntological operations consists of Carnapian-Goguenist *integration* of **o**ntologies. As stated previously, two **o**ntologies O_1 and O_2 are intuitively integrated into a third **o**ntology O, a reference **o**ntology, by means of re-interpreting or aligning O_1 and O_2 to suit the "point-of-view" (i.e. signature, sentences, models and semantics) of O.

While Lücke [235, p. 296] acknowledge the work of [342] on semantic integration, they present a less restrictive definition which is also based on refinements, and does not enforce the **o**ntology to be consistent regardless of the input **o**ntologies. This restriction lift permits the use of paraconsistent logics in all involved **o**ntologies (both input and reference). Their definition of ontological integration is as follows:

Definition 5.2.16 (Heterogeneous integration)**.** Given **o**ntologies O_1 and O_2 and a reference ontology O, in institutions I_1, I_2 and I, respectively, we say O *heterogenously (conservatively) integrates* O_1 and O_2 if there are (conservative) refinements from both O_1 and O_2.



In this diagram, O is the result of a heterogeneous integration between O_1 and O_2. If θ_1 and θ_2 are monomorphic, O is the result of a conservative heterogeneous integration.

Connection. As integration relies on the existence and a priori knowledge of a reference **o**ntology, it is often too difficult to guarantee that such an **o**ntology exists. For this reason, Lücke [235] propose ontological connection as a means of linking two **o**ntologies. This is achieved by specifying or computing a new **o**ntology over a so-called bridge theory formulated over a signature that crosses the two input **o**ntologies. They present three different possibilities of connection, with increasing levels of abstractness: connections through alignments, connections through interfaces and colimits and connections through specific modular **o**ntology languages.

The Carnapian-Goguenist approach to ontological connection is particularly suited to **o**ntologies built over a specific class of logics, namely description logics. For this reason, the terms "symbol" and "concept" are used nearly interchangeably in the sequel. That is because a symbol in the signature of a description logic (in institutional-theoretic lingo) is a concept.

Connections through alignments serve to address the issue of connecting two **o**ntologies by specifying an interface between them. This was initially presented by Zimmermann et al. [403] as the operation of *alignment*, of which there are two kinds: **V**-alignments and **W**-alignments (which are composed of **V**-alignments). Lücke [235] defines **V**-alignments as:

Definition 5.2.17 (**V**-alignments)**.** Given **o**ntologies O_1 and O_2, an *interface* for them is a triple

$$\langle \Sigma, \sigma_1 : \Sigma \to \mathsf{Sig}(O_1), \sigma_2 : \Sigma \to \mathsf{Sig}(O_2) \rangle$$

where $\mathsf{Sig}(O)$ denotes the signature of O's underlying institution. The interface informally specifies that:

- concepts $\sigma_1(c)$ in O_1 and $\sigma_2(c)$ in O_2 are identified for each concept c in Σ, regardless of whether the concepts share the same name and
- concepts in $\mathsf{Sig}(O_2) \setminus \bigcup_{c \in \mathsf{Sig}(O_1)} \sigma_1(c)$ and $\mathsf{Sig}(O_2) \setminus \bigcup_{c \in \mathsf{Sig}(O_2)} \sigma_2(c)$ are kept distinct, again regardless of whether the concepts share the same name.

The interface of an alignment does not provide the bridge **o**ntology required to connect the two **o**ntologies. To obtain it, one must compute the connection **o**ntology O through the interface (a process which Zimmermann et al. [403] call *merging*) by a category-theoretical pushout. The pushout is called a **V**-alignment over the interface.

Intuitively, the computed ontology O consists of a disjoint union of O_1 and O_2 with a way to identify what symbol belongs to what **o**ntology. It should be noted that once an interface has been established, computing the connection **o**ntology is a relatively straight-forward process. However, there is no unique way for automatically specifying the interface. Instead, this must be done manually.

Example 5.2.6. Consider two **o**ntologies $O_1 = \langle \{\text{Window}, \text{Bat}, \text{Job}\}, \Psi_1 \rangle$ and $O_2 = \langle \{\text{Window}, \text{Bat}, \text{Occupation}\}, \Psi_2 \rangle$, where Bat refers to baseball bat in O_1 but O_2 refers to the animal. Suppose an interface $\langle \{\text{Window}, \text{Job}\}, \sigma_1, \sigma_2 \rangle$ specifying that:

1. Window is mapped to the same symbol in O_1 and O_2, i.e. $\sigma_1(Window) = Window \in \mathsf{Sig}(O_1)$ and similarly for σ_2
2. Job is mapped to the same symbol in O_1 and to Occupation in O_2, i.e. $\sigma_2(Job) = Occupation \in \mathsf{Sig}(O_2)$

The **V**-alignment over the interface is constructed as the following pushout, where O is the connection **o**ntology as pictured in figure 5.2 below:

As Lücke [235] note, V-alignments are sufficient to deal with synonymy (a situation where different symbols have the same meaning, such as Job and Occupation in example

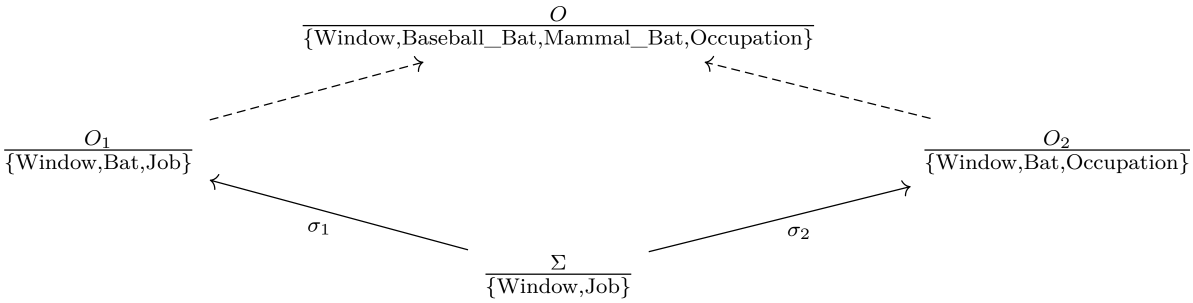


Figure 5.2: Example 5.2.6 visualized. Observe that σ_1 and σ_2 constitute a V-shape.

5.2.6) and homonymy (when identical symbols have different meanings, such as Bat in example 5.2.6), but they are still subject to issues of polysemy (when two symbols have different but related meanings). That is, given two **o**ntologies sharing the same symbol, a symbol is said to be *polysemous* if the theory of each **o**ntology contains different sentences concerning the same symbol. Once again taking the example 5.2.6, suppose that Window refers to the same concept but each input **o**ntology describes it via different axioms, thus Window is polysemous.

Additionally, **V**-alignments are not able to deal with concept subsumption — that is, a scenario where the input **o**ntologies encode concepts and subconcepts such as Person and Woman. Zimmermann et al. [403] present **W**-alignments to overcome the subsumption issue. The computed **o**ntology in a **W**-alignment is not the result of a pushout; rather, it is a a bridge **o**ntology. This bridge **o**ntology does not necessarily commit to the theories of the input **o**ntologies, but it includes subconcepts sentences for each subsumed concept, such as Woman $\sqsubseteq$ Person.

Furthermore, **M**-alignments are Lücke's answer to the issue of concept subsumption. They are a generalization of **V**-alignments whose computed **o**ntology do commit to the theories of each input **o**ntology [235].

Definition 5.2.18 (**M**-alignments)**.** Given two **o**ntologies O_1 and O_2, let $O_1^\sharp$ and $O_2^\sharp$ be (typically conservative) extensions of O_1 and O_2 respectively such that:

- new symbols are introduced to account for subsumed concepts
- new sentences are introduced for each subsumed concept to introduce subconcept relationships

An **M**-alignment over O_1 and O_2 is then defined as a **V**-alignment over $O_1^\sharp$ and $O_2^\sharp$. The connection **o**ntology O is computed as the pushout of the **M**-alignment. Figure 5.3 below contains an example of a **M**-alignment.

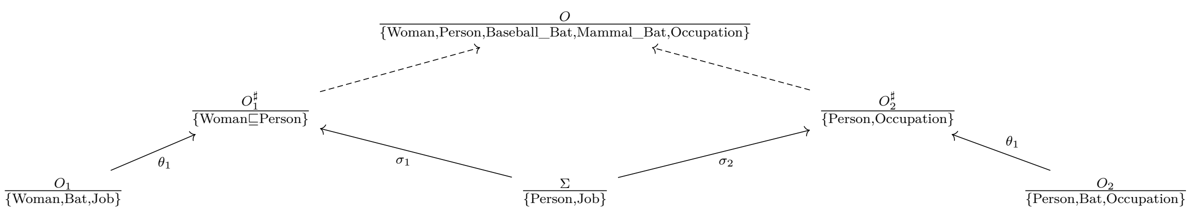


Figure 5.3: Example of **M**-alignment and pushout calculation.

Lücke [235] further generalize the concept of connection via an interface by calculating the colimit of a diagram (instead of a particular pushout). This is defined as follows.

Definition 5.2.19 (Colimit connection)**.** Given two **o**ntologies O_1 and O_2, take Σ_1 to be a sub-signature of O_1 and Σ_2 to be a sub-signature of O_1 and B an interface between O_1 and O_2 (in the sense of an alignment). The colimit connection of O_1 and O_2 is the ontology O computed from the diagram in figure 5.4.

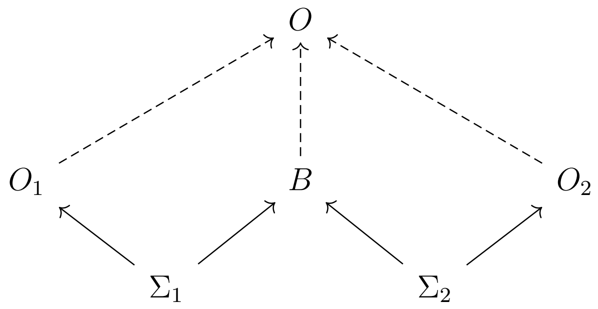


Figure 5.4: Colimit connection diagram.

Lastly, Lücke [235] present an approach to ontological connection based on distributed description logics (DDL) [43] and $\mathcal{E}$-connections [213]. Kutz introduced $\mathcal{E}$-connections in his doctorate thesis as a new technique for combining logics guaranteeing preservation of decidability — if the input logics are decidable, then so is their connection [211]. A detailed definition of ontological connection through $\mathcal{E}$-connections is beyond the scope of this section, as it is quite technically involved. However, what follows is an attempt to describe their intuitive working.

$\mathcal{E}$-connections may be thought of many-sorted heterogeneous theories where the component **o**ntologies can be formulated in different underlying logics, so long as they share the same many-sorted vocabulary. In order to relate the different **o**ntologies, $\mathcal{E}$-connections provide *link relations* and operators to manipulate and reason with link relations. In a certain way, $\mathcal{E}$-connections internalize the meta-discussion of connecting **o**ntologies and logics down to the language level — the act itself of connecting **o**ntologies is a language-level of object and thus can be manipulated. The reader can refer to [143, 144, 286] and, of course, [235] for an overview of ontological connections over $\mathcal{E}$-connections.

Some considerations

The Carnapian-Goguenist approach is appealing for a number of reasons. Primarily, the existing literature on institution theory is extensive, with Diaconescu's *Institution-Independent Model Theory* serving as a foundational text in the field of institution-free model theory — a "universal" model theory which does not depend on any particular underlying institution. Secondly, there is production-ready computer software based on this approach, the heterogeneous toolset (HETS) [265], supporting many **o**ntology representation languages and also automated reasoning through formal provers such as FaCT++ [383] and Isabelle [293]. HETS was originally introduced by Mossakowski as part of his habilitation thesis [261] and has since been continuously evolving.[9] In particular, it supports the representation language of Thousands of Problems for Theorem Provers (TPTP) system, a library for evaluating automated theorem provers, and thus also benefits from TPTP's connectivity to many provers. The current architecture of HETS as well as the languages it supports is pictured in figure 5.5, taken directly from the project's homepage.

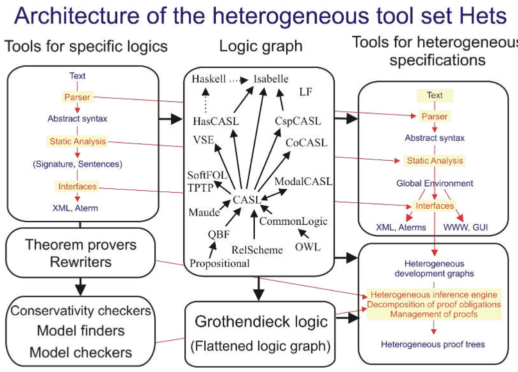


Figure 5.5: The architecture of HETS.

As previously mentioned, Carnapian-Goguenism also gave rise to a specification for ontological interoperability based on institution-powered formal semantics — DOL [263]. DOL is a specification language designed for knowledge representation and interoperability across different Ontology, Model and Specification (OMS) systems.

Despite the available literature and existing software, there are at least two clear shortcomings of Carnapian-Goguenism:

- Firstly, one must manage and specify logics based on their semantics. Institutions are restrictive in this sense, as the semantic entailment relation is specified as a family of *sets* restricted by *category-theoretic* models. This is prohibitive of using novel semantics, such as non-deterministic semantics [90] and possible translation semantics [245], at least without considerable effort when possible.
- In order to add a new kind of logic to an already existing development graph, one must *recompute* the Grothendieck institution from the graph. An alternative approach would be to assume that the Grothendieck institution must always be rebuilt. However this would entail a significant computational cost, which must be taken into account.

[9]Openly available at https://github.com/spechub/Hets.

Chapter 6

da Costian-Tarskianism

This chapter presents a novel approach for ontological heterogeneity that draws heavily from Carnapian-Goguenism, as described in the previous chapter. The approach is provisionally designated da Costian-Tarskianism, named after da Costa's Principle of Tolerance in Mathematics and after Alfred Tarski's work on the concept of a consequence operator. The steps to lay out our approach will follow exactly the same structure as section 5.2.1, in order to facilitate the comparison between da Costian-Tarskianism and Carnapian-Goguenism.

Tarski [380] initially presented the consequence operator as way to characterize what a logical system should preserve in order to be called a proper logic. The operator draws from the closure operator originally presented by Kazimierz Kuratowski, and it attempts to capture, as suggested by its name, the notion of what formulas should follow as syntactical consequences of a given theory in a rather general and abstract way. The consequence operator is defined as follows, as per Marcos [245, p. 17]:

Definition 6.0.1 (Consequence operator or relation)**.** For a given set For of formulas, $\Vdash \subseteq \mathcal{P}(For) \times For$ is a consequence operator (or relation) if the following three conditions hold:

1. If $A \in \Gamma$ then $\Gamma \Vdash A$ — reflexivity
2. If $\Delta \Vdash A$ and $\Delta \subseteq \Gamma$ then $\Gamma \Vdash A$ — monotonicity
3. If $\Delta \Vdash A$ and $\Gamma, A \Vdash B$ then $\Delta, \Gamma \Vdash B$ — transitivity

As previously noted, Marcos [245] renamed da Costa's Principle of Tolerance in Mathematics to Principle of Non-Triviality and formalized it using the consequence operator $\Vdash_{\mathbf{L}}$ over a given logic **L**. In addition to the Principle of Non-Triviality, they define two other logical principles:

Principle of Non-Contradiction (PNC)
L must be non-contradictory: $\exists\Gamma\forall A\ (\Gamma \nVdash_{\mathbf{L}} A$ or $\Gamma \nVdash_{\mathbf{L}} \neg A)$

Principle of Non-Triviality (PNT)
L must be non-trivial: $\exists\Gamma\exists B\ (\Gamma \nVdash_{\mathbf{L}} B)$

Principle of Explosion (PoE) or Principle of Pseudo-scotus (PPS)
L must be explosive: $\forall\Gamma, A, B\ (\Gamma, A, \neg A \Vdash_{\mathbf{L}} B)$

The objective of defining such principles is to capture meta-logical properties of a given logic. In this regard, da Costa's principle is in essence stating that no admissible logic should violate PNT. Marcos [245] defines, for instance, a paraconsistent logic as a logic such that PNC and PNT hold but not PPS. This meta-logical framework consists of the foundations of da Costa-Tarskianism.

6.1 What is a logic?

Unlike Carnapian-Goguenism, the da Costian-Tarskianist approach does not utilize institutions to define classes of logics via their *semantics*. Instead, consequence systems are chosen to represent classes of logics via their *syntax*. As W. Carnielli et al. [74] have shown, various kinds of calculi, such as Hilbert calculi, sequent calculi and tableau calculi, induce consequence systems. Consequently, consequence systems are sufficiently abstract to permit the construction of heterogeneous operations.

Prior to defining consequence systems, it is necessary to present some preliminaries. Many of the definitions hereafter are due to W. Carnielli et al. [74], the first of which is that of a signature.

Definition 6.1.1 (Signature)**.** A signature C is a countable family of sets C_k where $k \in \mathbb{N}$. The index k states the arity of the connectives, meaning C_k is the set of k-ary connectives. Usually C is a finite set.

The set C_0 of zero-ary connectives over a signature C is usually called the set of *constants* of C. Given two signatures C' and C'', it is customary to write $C' \leq C''$ if $C''_k \subseteq C'_k$ for every $k \in \mathbb{N}$. It may be noted that a signature C, in the current context, *does* contain the logical symbols.

A signature induces a language over it. Intuitively, the language of the signature is the set of all formulas that can be inductively constructed using its connectives and variables.

Definition 6.1.2 (Language of a signature)**.** Let C be a signature and $\Xi = \{\xi_n : n \in \mathbb{N}^+\}$ be an enumerable set called the set of *schema variables*. The language over C is the set $L(C)$ inductively defined as follows:

- $\xi \in L(C)$ for every $\xi \in \Xi$
- $c \in L(C)$ for every $c \in C_0$
- $(c(\phi_1, \ldots, \phi_k)) \in L(C)$ whenever $c \in C_k$, $k \geq 1$, and $\phi_1, \ldots, \phi_k \in L(C)$.

Signatures can be related in a categorial setting via signature morphisms. Signatures and signature morphisms constitute the category **Sig**. Observe that in this context, **Sig** denotes a *specific* and highly abstract category of signatures tailored for the approach, whereas the category **Sign** in definition 5.2.1 is specific to a particular institution.

Definition 6.1.3 (Signature morphism)**.** A *signature morphism* $h : C \to C'$ is a family of maps $h_k : C_k \to C'_k$, where $k \in \mathbb{N}$.

Definition 6.1.4 (Category of signatures)**.** The category of signatures **Sig** is defined as follows:

- The objects are signatures C
- The morphisms are signature morphisms as described above

Composition of morphisms h and g is defined as the composition of each indexed map, i.e. $h \circ g$ is the family of maps $h_k \circ g_k$ and the identity morphism is the family map of identity morphisms, i.e. id_k, where $k \in \mathbb{N}$.

One may generate a new signature from two inputs by means of fibring. The fibring of two signatures is defined below.

Definition 6.1.5. The *fibring of signatures* C' and C'' is the signature

$$C' \cup C''$$

such that $(C' \cup C'')_k = C'_k \cup C''_k$ for every $k \in \mathbb{N}$.

Signatures and their associated languages are fundamental to define consequence systems. As Citkin and Muravitsky [82] note, Tarski not only presented consequence operators in his 1930 paper but also consequence systems indirectly. Intuitively, consequence systems talk about the closure of consequence operators — that is, the set of all formulas which follow from a given theory. Consequence systems and consequence operators induce one another, as pointed out by Citkin and Muravitsky [82].

Definition 6.1.6 (Consequence System)**.** A *consequence system* is a pair $\langle C, \mathsf{C} \rangle$ where C is a signature and $\mathsf{C} : \mathcal{P}L(C) \to \mathcal{P}L(C)$ is a map with the following properties:

1. $\Gamma \subseteq \mathsf{C}(\Gamma)$ extensivity
2. $\Gamma_1 \subseteq \Gamma_2$ then $\mathsf{C}(\Gamma_1) \subseteq \mathsf{C}(\Gamma_2)$ monotonicity
3. $\mathsf{C}(\mathsf{C}(\Gamma)) \subseteq \mathsf{C}(\Gamma)$ idempotence

Before proceeding, it is necessary examine in greater detail the reasons for choosing consequence systems over more complex machinery from proof theory, such as abstract proof systems as presented by W. Carnielli et al. [74] themselves or even a categorial approach as described by Hyland [189]. As previously noted, consequence systems can be employed to represent the syntactical aspect of a wide array of logics. Indeed, consequence systems are sufficiently capable of expressing, intuitively, what set of sentences may be deduced from another set of sentences, but not precisely how. That is, the consequence operator does not encode a proof: the entire sequence of steps necessary to deduce a set of consequents from a set of antecedents. While this may be a shortcoming for certain applications where explainability is a major concern, it is not the case from the ontological point of view. The primary objective of a heterogeneous applied ontological framework is to provide the ability to describe and reason about concepts. However, the specific mechanisms underlying reasoning are relegated to the domains of metaphysics and grounding. The primary contention of this paper is that, in the field of applied **O**ntology, the **o**ntologies do not require self-awareness, that is, they do not need to encode information about their own inner workings.

Consequence systems represent a *class* of logics and are devoid of a semantic construct, thus exhibiting a sort of duality to institutions. A consequence system may also be attached to a semantic construct, such as algebraic semantics as done via W. Carnielli et al.'s interpretation systems [74, p. 92], resulting in a signature-**bound** logic (also called a logic system). Because consequence systems are built on top of a *particular* signature, this is a key distinction between them and institutions. This fact is also the reason the duality is not mirror-like. As noted by M. Coniglio [89], a better candidate for proper mirror-like duality, when compared to institutions, are Meseguer's entailment systems.

W. Carnielli et al. [74] also define some particular types of consequence systems. In what follows, $\mathcal{P}_{\mathsf{fin}}S$ denotes the set of all finite subsets of S.

Definition 6.1.7 (Compact Consequence System)**.** A consequence system $\langle C, \mathsf{C}\rangle$ is *compact* or *finitary* if

$$\mathsf{C}(\Gamma) = \bigcup_{\Phi \in \mathcal{P}_{\mathsf{fin}}\Gamma} \mathsf{C}(\Phi)$$

For each $\Gamma \subseteq L(C)$. Compact consequence systems are also called *abstract systems* in [108].

Definition 6.1.8 (Quasi-consequence System)**.** A *quasi-consequence system* is a consequence system such that idempotence, as defined in 6.1.6 does not necessarily hold.

Definition 6.1.9. A consequence system $\langle C, \mathsf{C}\rangle$ is *closed for renaming substitutions* if

$$\sigma(\mathsf{C}(\Gamma)) \subseteq \mathsf{C}(\sigma(\Gamma))$$

for every $\Gamma \in L(C)$ and every renaming substitution σ, i.e. $\sigma(\{\zeta\}) = \{\zeta'\}$ for each $\zeta \in \Xi$, where $\zeta' \in \Xi$. If the inclusion holds for every substitution, the consequence system is called *structural*.

Consequence systems may be related in an order theory sense, by introducing a weakness relation, and also in a categorial setting, by defining morphisms between consequence systems. Consequence systems and their morphisms constitute a category, **Csy**.

Definition 6.1.10 (Weakness relation)**.** The consequence system $\langle C, \mathsf{C}\rangle$ is *weaker* than consequence system $\langle C', \mathsf{C}'\rangle$, written

$$\langle C, \mathsf{C}\rangle \leq \langle C', \mathsf{C}'\rangle$$

if $L(C) \subseteq L(C')$ and $\mathsf{C}(\Gamma) \subseteq \mathsf{C}'(\Gamma)$ for every subset Γ of $L(C)$. Additionally, $\langle C, \mathsf{C}\rangle$ is *partially weaker* than consequence system $\langle C', \mathsf{C}'\rangle$, written

$$\langle C, \mathsf{C}\rangle \leq_p \langle C', \mathsf{C}'\rangle$$

if $L(C) \subseteq L(C')$ and $\mathsf{C}(\emptyset) \subseteq \mathsf{C}'(\emptyset)$.

Definition 6.1.11 (Consequence system morphisms)**.** A *consequence system morphism* $h : \langle C, \mathsf{C}\rangle \to \langle C', \mathsf{C}'\rangle$ is a map $h : L(C) \to L(C')$ such that

$$h(\mathsf{C}(\Gamma)) \subseteq \mathsf{C}'(h(\Gamma))$$

for every $\Gamma \subseteq L(C)$.

One can think of consequence system morphisms as transformations that weakly translate consequence. If $\{\phi\}$ is a consequent of Γ via C in the source consequence system, then $h(\{\phi\})$ is a consequent of $h(\Gamma)$ in the target consequence system.

Definition 6.1.12 (Category of consequence systems)**.** The category of consequence systems **Csy** is defined as follows:

- The objects are consequence systems $\langle C, \mathsf{C}\rangle$
- The morphisms are consequence system morphisms as described above

Composition of morphisms h and g is defined as the usual composition of maps $g \circ h$. The identity morphism is the identity map, i.e. $h(X) = X$ for every X.

The category **Csy** may be thought of as the web which connects classes of logics (i.e. consequence systems) whenever it is possible to establish a weak translation between their deductive aspect. That is, if a consequence system's consequence map C is a subset of another consequence system, under some translation, then there is a morphism between them.

6.2 What is an ontology?

SIMILARLY to institutions, consequence systems, in and of themselves, are insufficient to describe **o**ntologies, as they lack the mechanisms to encode ontological information[1]. For this reason, it is necessary to employ an approach analogous to the Carnapian-Goguenist way, whereby the logical machinery is extended with ontological axioms.

The intuitive understanding behind this approach is that an **o**ntology is merely a consequence system plus a theory (i.e. the set of ontological axioms). This method of definining **o**ntologies is more straight-forward than the Carnapian-Goguenist approach, although it initially places **o**ntologies in an **o**ntological vacuum (as in chapter 2). extended development graphs represent a potential solution to this apparent limitation, as they permit **o**ntologies to be related.

It is necessary to note that one could "append" axioms, representing an ontological theory, to an existing consequence system by changing its consequence operator. By doing so, one obtains a new class of logics including the ontological theory as axioms. Thus, a consequence system could be defined as an **o**ntology itself. However, the reasons for not doing so are of philosophical and computational nature. On the first point, by making the ontological theory explicit, one draws the boundary between what is inherent to an intuitive "mode of reasoning" (i.e. a class of logics, the consequence system itself) and what is particular to a description of the world (i.e. the ontological axioms themselves). This position follows N. C. A. d. Costa's view on the distinction between Logic and Ontology, both as philosophical areas and as objects, as exposed in [97]. Furthermore, such position is aligned with Guarino's thesis that domain and reasoning knowledge should be independent Guarino [158]. With regard to the second point, clearly specifying the

[1]This claim relies on the philosophical distinction between **O**ntology and Logic, as described in chapter 2

ontological axioms, and presenting an algorithmic way to manage such axioms, increases the inherent modularity of the approach. By not equating a consequence system to an **o**ntology, it is possible to utilize a single consequence systems to generate several **o**ntologies based on it.

As a preliminary step, below is the definition of an extended consequence system.

Definition 6.2.1 (extended consequence system)**.** An *extended consequence system* is a quadruple $\langle C, \mathsf{C}, C_o, \Gamma_o \rangle$, where C, C_o are signatures, C is a consequence map and $\Gamma_o \in L(C)$ is a set, such that:

1. $\langle C, \mathsf{C} \rangle$ is a consequence system
2. For every $C_k \in C$ and $C'_k \in C_o$, $C'_k \subset C_k$
3. For every $\phi \in \Gamma_o$, $\phi \in \mathsf{C}(\emptyset)$ — i.e. Γ_o is an axiomatic theory

An **o**ntology is then defined as a *particular kind* of extended consequence system.

Definition 6.2.2 (**O**ntology)**.** An ***o**ntology* is defined as a particular extended consequence system $\langle C, \mathsf{C}, C_o, \Gamma_o \rangle$. When the underlying $\langle C, \mathsf{C} \rangle$ consequence system is implicitly understood, we may drop it and refer to an **o**ntology by its components C_o and Γ_o (also called ontological aspect components).

The rationale behind extending a consequence system is now evident. It is possible to distinguish between the ontological content and the purely logical machinery, while allowing the machinery to handle ontological knowledge. From this point onward, the terms "extended consequence system" and **o**ntology are used interchangeably in the context of the da Costa-Tarski approach. The point is that **o**ntologies are a specific kind of extended consequence systems whose theory Γ is defined to match the definition of **o**ntology discussed previously in definition 2.2.11 by Guarino.

Extended consequence systems and their morphisms constitute the category **ECsy**:

Definition 6.2.3 (Category of extended consequence systems (or **o**ntologies))**.** The category **ECsy** of extended consequence systems is defined as follows:

- Its objects are extended consequence systems $\langle C, \mathsf{C}, C_o, \Gamma_o \rangle$
- A morphism between extended consequence systems $\langle C^A, \mathsf{C}^A, C_o^A, \Gamma^A \rangle$ and $\langle C^B, \mathsf{C}^B, C_o^B, \Gamma^B \rangle$ is a consequence system morphism $h : \langle C^A, \mathsf{C}^A \rangle \to \langle C^B, \mathsf{C}^B \rangle$ such that $h(\Gamma^A) = \Gamma^B$.

Composition between extended consequence system morphisms works as expected, composing consequence system morphisms and composing the Γ mappings. The identity morphism is the consequence system identity morphism, since $id(\Gamma) = \Gamma$.

It should be noted that **o**ntologies are not defined with respect to development graphs and do not require a Grothendieck construction in order to relate two **o**ntologies based on different underlying logics. Nevertheless, it is possible to construct a graph structure that allows **o**ntologies to be related via morphisms of **ECsy**, and additionally via other operations such as fibring and splitting. Consequently, **o**ntologies in the da Costa-Tarski sense are sufficiently flexible to be independent of external structures such as graphs. However, they permit the existence of such external structures in a manner that is both useful and coherent.

6.3 How to relate ontologies?

This section details the structures and mechanisms required to relate **o**ntologies. The following is based on the machinery developed by W. Carnielli et al. [74] to reinterpret morphisms and other operations between consequence systems as means to refine, integrate and connect **o**ntologies.

Yet again inspired by the approach from Lücke [235], we shall now define the concept of an extended development graph. This will form the basis of our framework and will slowly become mutated with add-ons to increase complexity and expressivity of operations.

Definition 6.3.1. An *extended development graph* is a vertex and edge-labeled directed, acyclic graph (DAG) $\mathcal{AG} = \langle \mathcal{N}, \mathcal{L} \rangle$, where:

- $\mathcal{N}$ is a set of nodes, where each node $N \in \mathcal{N}$ is labeled with an extended consequence system $\langle C^N, \mathsf{C}^N, C_o^N, \Gamma_o^N \rangle$
- $\mathcal{L}$ is a sorted set of directed links:
  - *definition links* $K \xrightarrow{h} N$ where h is an extended consequence system morphism
  - *theorem links* $K \overset{t}{\dashrightarrow} N$ for each $K, N \in \mathcal{N}$ if the consequence system labeling K is weaker than the consequence system labeling N, c.f. definition 6.1.10.

An extended development graph captures a web of **o**ntologies related by definition links, corresponding to morphisms, and induced links, corresponding to existing weakness relations. If there is a theorem link between two **o**ntologies, it is possible to informally assess that one is a non-conservative sub-**o**ntology of the other. Same as in the Carnapian-Goguenist approach, the fact extended development graphs are directed and acyclic ensures that there are no circular definitions.

6.3.1 Refinement

Ontologies defined via extended consequence systems can also be refined in a manner that is highly analogous to the process of refining their institution-based counterpart. There are two kinds of refinements, homogeneous and heterogeneous.

Definition 6.3.2 (Homogeneous refinements)**.** Given two **o**ntologies O_1 and O_2 in an extended development graph, O_2 is called a *homogeneous refinement* of O_1 if there is a theorem link $O_1 \overset{t}{\dashrightarrow} O_2$.

Definition 6.3.3 (Heterogeneous refinements)**.** Given two **o**ntologies O_1 and O_2, O_2 is a *heterogeneous refinement* of O_1 in the underlying extended development graph if:

- there is a third **o**ntology O_2' such that a definition link $O_1 \xrightarrow{h} O_2$ exists, where h is monomorphic
- there is a theorem link $O_1 \overset{t}{\dashrightarrow} O_2$

Refinements in the context of extended consequence systems are sufficiently similar to those of institutions that they may be represented diagrammatically in the same manner, as depicted in figure 6.1. However, the meaning of each link is quite different in each approach. In the Carnapian-Goguenist approach, for one ontology to be heterogeneously refined into another, the latter must conserve models of the former. On the other hand, in the da Costian-Tarskian approach, refinement conserves theoremhood.

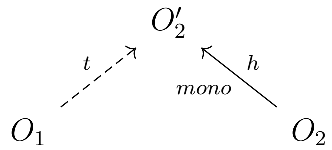


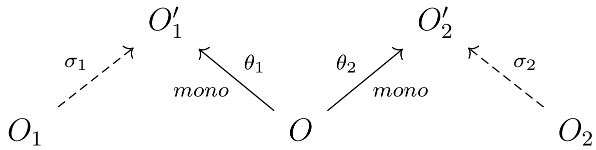
Figure 6.1: Heterogeneous refinement from O_1 to O_2.

One can also informally define mirrored definitions of equivalence relations, as presented in definitions 5.2.14 and 5.2.15. Two ontologies are weakly equivalent if they can be heterogeneously and conservatively refined into each other (i.e. they preserve each other's theorems), and they are pre-equivalent if they share a common monomorphic extension.

6.3.2 Integration

As defined by Lücke [235], integration between ontologies is based on refinements. This approach will be followed here, as it is a parallel to the existing framework.

Definition 6.3.4 (Heterogeneous integration)**.** Let O_1 and O_2 be ontologies and O a so-called reference ontology, in a given extended development graph. We say O *heterogenously (conservatively) integrates* O_1 and O_2 if there are (conservative) refinements from both O_1 and O_2.



In this diagram, O is the result of a heterogeneous integration between O_1 and O_2. If θ_1 and θ_2 are monomorphic, O is the result of a conservative heterogeneous integration.

6.3.3 Connection

Thus far, it appears that extended consequence systems and institutions are operationally analogous, as evidenced by the superficial similarity of the machinery for connecting and integrating. From this point onward, the operational parallels will be broken down by the introduction of a new tooling system. As a brief reminder, the objective of the connection operation between two ontologies is to generate a third, novel ontology that retains characteristics of the input ontologies. In the da Costa-Tarskian approach, ontologies may be connected by means of *fibring*.

Fibring was originally presented by Gabbay as a means to combine two normal modal logics into a third normal bimodal logic [127, 128]. A. Sernadas, C. Sernadas, and Caleiro [344] generalized fibring to deal with propositional logics in general — given two propositional logics L_1 and L_2, the fibring of them is a new logic $L_1 \circ L_2$ which is a minimal conservative extension of both. Although fibring has been further generalized in various accounts (see, for instance, [61, 346, 349]), we will present a slightly modified version of algebraic fibring as presented in [74, p. 150-160] for extended consequence systems.

The intuitive idea behind fibring is it joins two logics by translating them into a language guaranteed to be conflict-free. As a matter of fact, before formally defining the fibring operation it is necessary to define translations between languages of consequence systems.

Definition 6.3.5 (Translations and substitutions)**.** Let C and C' be two signatures such that $C \leq C'$ and $g : L(C') \to \mathbb{N}$ a bijection. The *translation*

$$\tau_g : L(C') \to L(C)$$

is the map defined inductively as follows:

- $\tau_g(\xi_i) = \xi_{2i+1}$ for $\xi_i \in \Xi$
- $\tau_g(c) = c$ for $c \in C_0$
- $\tau_g(c') = \xi_{2g(c')}$ for $c' \in C'_0 \setminus C_0$
- $\tau_g(c(\gamma'_1, \ldots, \gamma'_k)) = (c(\tau_g(\gamma'_1), \ldots, \tau_g(\gamma'_k)))$ for $c \in C_k$, $k > 0$ and $\gamma'_1 \ldots, \gamma'_k \in L(C')$
- $\tau_g(c'(\gamma'_1, \ldots, \gamma'_k)) = \xi_{2g(c'(\gamma'_1,\ldots,\gamma'_k))}$ for $c' \in C'_k \setminus C_k$, $k > 0$ and formulas $\gamma'_1, \ldots, \gamma'_k \in L(C')$

The preliminary *substitution*

$$\tau_g^{-1} : \Xi \to L(C')$$

is defined as

- $\tau_g^{-1}(\xi_{2i+1}) = \xi_i$ for $\xi_i \in \Xi$
- $\tau_g^{-1}(\xi_{2i}) = g^{-1}(i)$

If a preliminary substitution τ_g^{-1} is extended to $L(C)$, corresponding to the proper inverse of τ_g, it is called a proper substitution or simply a substitution[2].

Given a variable in $\tau_g(L(C'))$, one can determine if it comes from a variable or a formula starting with a connective in $C' \backslash C$ if its index is even or odd. Additionally, it may be noted that the translation τ_g maps symbols of C' that are not in C to propositional variables. This is by design, to ensure that no symbols of C' get mapped to existing symbols of C.

[2]In order to extend a preliminary substitution to a proper substitution, it is necessary to present the inductive rules for the values of τ_g^{-1} and prove that $\tau_g \circ \tau_g^{-1} = \tau_g \circ \tau_g^{-1} = id$. The details of this construction are present in [74, p. 150-151].

In order to define fibring of consequence systems $\langle C', \mathsf{C}'\rangle$ and $\langle C', \mathsf{C}''\rangle$, it is necessary to define translations between $L(C' \cup C'')$ and the input languages, $L(C')$ and $L(C'')$. Given a bijection $g : L(C \cup C') \to \mathbb{N}$, such translations are defined as follows:

$$\tau'_g : L(C' \cup C'') \to L(C') \text{ and } \tau''_g : L(C' \cup C'') \to L(C'')$$

The corresponding substitutions are

$$\tau'^{-1}_g : L(C') \to L(C' \cup C'') \text{ and } \tau''^{-1}_g : L(C'') \to L(C' \cup C'')$$

For the sake of readability in what follows, a few provisional measures will be taken. The suffix g will be dropped for translations and substitutions by assuming that a fixed bijection $L(C' \cup C'') \to \mathbb{N}$ exists. Given a set $\Gamma \in L(C)$, $\tau(\Gamma)$ denotes the following set:

$$\tau(\Gamma) = \{\tau(\phi) : \phi \in \Gamma\}$$

Additionally, in order to minimize confusion between the signature of a consequence system and the consequence system itself, consequence maps will be denoted by $\vdash$ at times. In this case, given a consequence system $\langle C, \vdash\rangle$ and set $\Gamma \in L(C)$, $\Gamma^{\vdash}$ denotes $\vdash(\Gamma)$ and is called the *deductive closure* of Γ.

Given two consequence systems $\mathfrak{C}'$ and $\mathfrak{C}''$, translations and substitutions define the closure of each $L \subseteq L(C' \cup C'')$ with respect to the consequence maps of each consequence system.

Definition 6.3.6. Let $\mathfrak{C}' = \langle C', \vdash'\rangle$ and $\mathfrak{C}'' = \langle C'', \vdash''\rangle$ be two consequence systems and let $\Gamma \subseteq L(C' \cup C'')$. Assume that $\tau' : L(C) \to L(C')$ and $\tau'' : L(C') \to L(C)$ denote the translations between the two consequence systems. Similarly, τ'^{-1} and τ''^{-1} denote the respective substitutions between the consequence systems. The $\vdash'$-*closure* of Γ is the set:

$$\Gamma^{\vdash'} = \tau'^{-1}((\tau'(\Gamma))^{\vdash'})$$

Similarly, the $\vdash''$-*closure* of Γ is the set $\Gamma^{\vdash''} = \tau''^{-1}((\tau''(\Gamma))^{\vdash''})$

Intuitively, the $\vdash'$-closure of a set $\Gamma \in L(C' \cup C'')$ denotes the set obtained by first translating Γ to the language of consequence system $\mathfrak{C}'$, computing its deductive closure, and mapping it back to the language $L(C' \cup C'')$.

Finally, the fibring of consequence systems is defined as follows.

Definition 6.3.7. Let $\mathfrak{C}' = \langle C', \vdash'\rangle$ and $\mathfrak{C}'' = \langle C'', \vdash''\rangle$ be two consequence systems. The *fibring of two consequence systems* $\mathcal{C}'$ and $\mathcal{C}''$ is a pair

$$\mathcal{C}' \cup \mathcal{C}'' = \langle C, \vdash\rangle$$

where

- C is a signature such that $C_k = C'_k \cup C''_k$ for every $k \in \mathbb{N}$
- $\vdash: \mathcal{P}(L(C)) \to \mathcal{P}(L(C))$ where, for each $\Gamma \subseteq L(C)$, $\Gamma^{\vdash}$ is inductively defined as follows:

1. $\Gamma \subseteq \Gamma^{\vdash}$
2. If $\Delta \subseteq \Gamma^{\vdash}$, then $\Delta^{\vdash'} \cup \Delta^{\vdash''} \subseteq \Gamma^{\vdash}$

The object generated by fibring two consequence systems is also a consequence system, as proven by By W. Carnielli et al. [74, prop. 4.1.24]. One may conceptualize the resulting consequence system as being a disjoint union of the two input consequence systems, where their respective consequence maps are preserved, albeit under translations and substitutions, via the deductive closure.

It is important to note that consequence systems are not **o**ntologies in the current context. Consequently, it is necessary to extend the definition of fibring to extended consequence systems. This will provide a means to in fact to connect **o**ntologies.

Definition 6.3.8 (Heterogeneous connection)**.** Let $O_1 = \langle C^1, \vdash^1, C_o^1, \Gamma_o^1 \rangle$ and $O_2 = \langle C^2, \vdash^2, C_o^2, \Gamma_o^2 \rangle$ be two extended consequence systems, i.e. **o**ntologies. A tuple $O_1 \cup O_2 = \langle C, \vdash, C_o, \Gamma_o \rangle$ is the *connection* of O_1 and O_2 if:

- $\langle C, \vdash \rangle$ is the result of fibring between $\langle C^1, \vdash^1 \rangle$ and $\langle C^2, \vdash^2 \rangle$
- $C_o = C_o^1 \cup C_o^2$
- $\tau_1^{-1}(\phi) \cup \tau_2^{-1}(\psi) \subseteq \Gamma_o$ for each $\phi \in \Gamma_o^1, \psi \in \Gamma_o^2$

where τ_1^{-1} and τ_2^{-1} are the substitutions derived via fibring.

The connection between two **o**ntologies constructs an object whose consequence system aspect is the result of the fibring of the consequence systems in the input **o**ntologies, and whose ontological aspect is preserved under substitution. We shall now prove the resulting object is indeed an **o**ntology.

Proposition 6.3.1. *The heterogeneous connection $O_1 \cup O_2$ of **o**ntologies O_1 and O_2 is an **o**ntology.*

Proof. Let $O_1 = \langle C^1, \vdash^1, C_o^1, \Gamma_o^1 \rangle$ and $O_2 = \langle C^2, \vdash^2, C_o^2, \Gamma_o^2 \rangle$. Suppose $O_1 \cup O_2 = \langle C, \vdash, C_o, \Gamma_o \rangle$. By W. Carnielli et al. [74, prop. 4.1.24], $\langle C, \vdash \rangle$ is a consequence system. By definition, C_o is a signature and it is straight-forward to see that $C_o \subseteq C$.

Consider $\emptyset^{\vdash}$ obtained via fibring, i.e the axiomatic theory of $\langle C, \vdash \rangle$. Clearly, by definition, we have that $\emptyset \in \emptyset^{\vdash}$. Therefore $\emptyset^{\vdash^1} \cup \emptyset^{\vdash^2} \subseteq \emptyset^{\vdash}$.

Note that for each $\phi \in \Gamma_o^1$, $\tau_1^{-1}(\phi) \in \emptyset^{\vdash^1} \subseteq \emptyset^{\vdash}$. Similarly, for every $\psi \in \Gamma_o^2$, $\tau_2^{-1}(\psi) \in \emptyset^{\vdash^2} \subseteq \emptyset^{\vdash}$. Thus, $O_1 \cup O_2$ conserves the ontological aspect and therefore is an **o**ntology. □

The ontological aspect of the resulting connected **o**ntology naively merges the ontological aspects of the input **o**ntologies. That is, no additional steps are taken to handle the aforementioned issues of synonymy (a situation where different symbols have the same intended meaning) and homonymy (a situation where the same symbol has different meanins). Rather, these issues can be dealt with via morphisms in **ECsy**. Recall the fact **o**ntologies may live inside an extended development graph, then heteregeneous

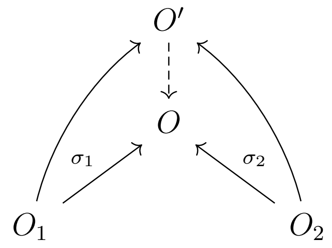


Figure 6.2: Heterogeneous connection of O_1 and O_2.

connection can be interpreted as adding the coproduct of O_1 and O_2 into the graph alongside the required morphisms, as visualized in figure 6.2.

If two **o**ntologies share the same common vocabulary, i.e. the signature of the ontological aspect shares common symbols, then one may add morphisms to "prepare" the input **o**ntologies. This sort of "preparation" via extra morphisms allows one to ensure the input **o**ntologies refer to the same concepts using the same symbols and use different symbols when there is a homonymy issue. This is depicted in figure 6.3.

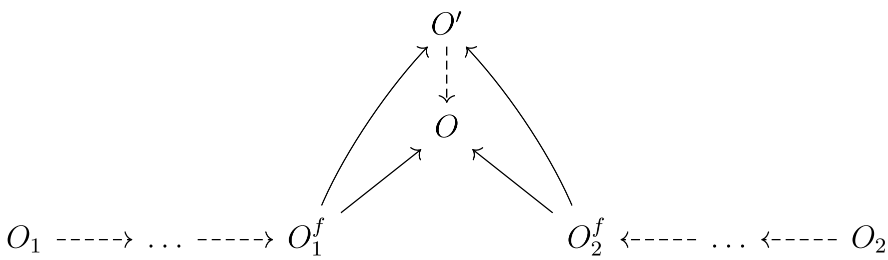


Figure 6.3: Heterogeneous connection of O_1 and O_2, with "preparation". Note the original input **o**ntologies may pass through many steps prior to connecting, resulting in final **o**ntologies O_1^f and O_2^f.

Ontological connection via fibring is a compelling concept, as it offers a means of combining ontologies while ensuring a minimal extension and preserving certain properties. Results compiled by W. Carnielli et al. [74, p. 150-160] include preservation of structurality, weakness relation transitivity when the consequence systems of input **o**ntologies are structural and compactness. Additionally, they present three characteristics which increase the attractiveness of fibring: firstly, it consists of a homogeneous combination mechanism; secondly, the mechanism is algorithmic and easily extendable, as it has been done for extended consequence systems; thirdly, the resulting combination is canonical in that it is minimally stronger than the input consequence systems of the **o**ntologies.

Fibring-based ontological connection has a few clear shortcomings. As shown by Marcelino, Caleiro, and Baltazar in a series of papers [60, 242, 243, 244], fibring in general does not preserve logical decidability — a very sought-after property for computational purposes. However, disjoint fibring (when the input consequence systems have disjoint signatures) does preserve decidability. Additionally, Marcelino, Caleiro, and Baltazar [244] proved that, when decidability is preserved in the fibred consequence system, the complexity of the decision problem can be polynomially reduced to the worst decision problem of the input consequence systems.

As previously stated, consequence systems may be enriched with interpretation systems to construct signature-bound *logic systems*. In addition, logic systems may also be fibred, which raises the question of the status of soundness and completeness preservation. A. Sernadas, C. Sernadas, and Zanardo [347] and Zanardo, A. Sernadas, and C. Sernadas [401] investigate under which conditions fibring preserves completeness, by showing classes of complete logic systems whose fibring is also complete.

Fibring, more specifically algebraic fibring, is far from the only kind of combination operation in the literature applicable to consequence systems, logic systems and logics in general. It is not in scope of this section to discuss in great detail about other operations, as its purpose is to establish an initial ground upon which to build heterogeneous **o**ntologies based on consequence systems. However, what follows are very brief overviews of a few of these works, as they address property-preservation issues and also illustrate potential paths for further development of the da Costian-Tarskianist approach:

1. *Graph-theoretic fibring*: A. Sernadas et al. [346] and A. Sernadas et al. [348] present a graph-theoretic account of fibring based on multi-graphs (or *m-graphs*) targetting the point-wise combination of the models of input logic systems. It has been shown by M. E. Coniglio, A. Sernadas, and C. Sernadas [88] that this kind of fibring preserves the finite model property, which entails decidability under certain conditions (see [230] for more on this matter).

2. *Importing logics*: an asymmetric combination technique originally devised by Rasga, A. Sernadas, and C. Sernadas [319] where the combined logic system is endowed with an importing connective, which allows formulas of the imported logic to be transformed into formulas of the importing logic. Rasga, A. Sernadas, and C. Sernadas [318] have shown this technique does preserve soundness and completeness and in [317] it is shown fibring may be characterized by *biporting*, an extension of importing which includes an exporting connective.

3. *Meet-combination of logics*: originally devised by A. Sernadas, C. Sernadas, and Rasga [345], meet-combination is an approach to connecting logics based on combined constructors, which generate melded connectives inheriting the intersection of the properties of the input logic instead of their conjunction. Meet-combination was shown to preserve completeness and decidability under a set of admissable rules in [320].

6.3.4 Decomposition

The intuitive idea of decomposition is to extract one or more sub-**o**ntologies from a given **o**ntology, in such a way that the sub-**o**ntologies are weaker, in a sense, than the original **o**ntology. The concept of ontological decomposition requires extending the definitions of **o**ntology and development graph to include the splitting machinery presented in W. Carnielli et al. [74, p. 391-400].

The initial step towards decomposition is the notion of a k-restricted language.

Definition 6.3.9 (k-restricted language)**.** Given $k \in \mathbb{N}$ and a signature C, $L(C)[k]$ is the set of formulas ϕ in $L(C)$ such that the set of schema variables occurring in ϕ is exactly $\{\xi_1, \dots, \xi_k\}$.

The category **sSig** which extends **Sig** by introducing splitting signature morphisms, based on k-restricted languages.

Definition 6.3.10 (Splitting signature morphism)**.** Given two signatures C and C', a *splitting signature morphism* f between them, denoted $f : C \to C'$, is a mapping $f : L(C) \to L(C')$ such that if $c \in C_k$ then $f(c) \in L(C')[k]$.

A splitting signature morphism f induces a mapping $\hat{f} : L(C) \to L(C')$ such that:

- $\hat{f}(\xi) = \xi$ if $\xi \in \Xi$;
- $\hat{f}(c) = f(c)$ if $c \in C_0$;
- $\hat{f}(c(\phi_1, \ldots, \phi_k)) = f(c)(\hat{f}(\phi_1), \ldots, \hat{f}(\phi_k))$ if $c \in C_k$, $\phi_1, \ldots, \phi_k \in L(C)$ and $k > 0$.

The two categories **sSig**, the category of splitting signatures, and **sCon**, the category of splitting consequence systems, provide the necessary basis for ontological decomposition.

Definition 6.3.11 (Category **sSig**)**.** The category **sSig** is defined as follows:

- Its objects are signatures;
- Its morphisms are splitting signature morphisms.

Composition between two signature morphisms $f : C \to C'$ and $g : C' \to C''$, denoted by $g \cdot f$, is defined to be the signature morphism $g \cdot f : C \to C''$ given by the mapping $\hat{g} \circ f : |C| \to L(C'')$.

The identity morphism $id_C : C \to C$ is defined as $id_C(c) = c$ for $c \in C_0$ and $id_C(c) = c(\xi_1, \ldots, x_k)$ for $c \in C_k$, $k > 0$.

Definition 6.3.12 (Category **sCon**)**.** The category **sCon** of splitting consequence systems is defined as follows:

- Its objects are consequence systems;
- A morphism $f : \mathcal{C} \to \mathcal{C}'$ is a morphism $f : C \to C'$ in **sSig** such that, for every $\Gamma \cup \{\phi\} \subseteq L(C)$,
$$\Gamma \vdash \phi \text{ implies } \hat{f}(\Gamma) \vdash' \hat{f}(\phi)$$

Composition and identity morphisms are as in **sSig**.

It is not necessary to extend the **sCon** into an analogous category whose objects are extended consequence systems for a few reasons. The morphisms in **sCon**, by definition, preserve entailment over a splitting signature morphism, which is necessary requirement in the context of ontological knowledge. Furthermore, the operations of ontological refinement, integration and connection insist on preserving or adding onto the ontological aspect of extended consequence systems — it does not make sense to impose this restriction in the case of ontological decomposition, since the intuitive goal is to be able to extract significant chunks of an **o**ntology into sub-**o**ntologies.

Before formally defining ontological decomposition, the following proposition is crucial to guarantee the machinery we use does indeed produce **o**ntologies.

Proposition 6.3.2. *The category* **sCon** *has products of arbitrary small, non-empty families of objects. Moreover, if every object of the family is structural, so is the product.*

Proof. Due to W. Carnielli et al. [74, prop. 9.1.6]. □

In order to define ontological decomposition, one needs to extend the extended development graph to include extra links tied to splitting consequence system morphisms. This new structure is called an extended splittable development graph, henceforth ESDG, and it is defined as follows.

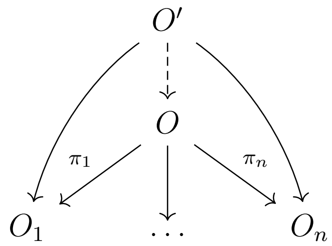
Definition 6.3.13 (extended splittable development graph)**.** An *extended splittable development graph* or ESDG is an extended development graph $\mathcal{AG} = \langle \mathcal{N}, \mathcal{L} \rangle$ such that $\mathcal{L}$ contains additional directed links:

- *splitting links* $K \xrightarrow{f} N$ for each K, N such that there exists a morphism f in **sCon** between the underlying consequence systems labelling K and N

Finally, by adding splitting links, one can define ontological decomposition.

Definition 6.3.14 (Ontological decomposition)**.** Let $O = \langle C, \vdash, C_o, \Gamma_o \rangle$ be an **o**ntology in an ESDG. We say O *decomposes into a family of* ***o****ntologies* $O_1, \ldots, O_n$ if:

1. there are splitting links from O to each O_i, $1 \leq i \leq n$;
2. the generated diagram is a product.



Observe that, as previously stated, decomposition does not impose restrictions on the ontological aspect and instead relies on entailment preservation provided by splitting morphisms. Perhaps non-intuitively, decomposition also does not enforce the "component" **o**ntologies to be weaker than the input **o**ntology in any sense. It is often times desirable to decompose into a weaker family of **o**ntologies, however it is not necessary to ensure at minimum structurality as per proposition 6.3.2.

Ontological decomposition is also a means to provide possible-translation characterization and semantics to **o**ntologies, in the sense of W. Carnielli et al. [74]. As the current objective is to present an initial proposal of ontological decomposition, it is not in scope to address this topic in depth at this time. Nevertheless, it is possible that ontological decomposition could result in the development of interesting ontological machinery. Indeed, some **o**ntologies, in practice, already implement some form of decomposition as a means of describing a target **o**ntology. For example, e.g. BFO is decomposed into its SNAP and SPAN sub-**o**ntologies.

6.4 Some considerations

At this stage, da Costian-Tarskianism is merely a proposal, for there is no practical, ready-to-use, production-ready implementation of it similar to Carnapian-Goguenism's Hets. Notwithstanding, it is possible to implement the approach using a computational category-theoretical framework, such as the one currently being developed by the AlgebraicJulia[3] project, which in turn is based on the work developed by Patterson, Lynch, and Fairbanks [291] on attributed $\mathcal{C}$-sets.

Despite lack of implementation, da Costian-Tarskianism has a few theoretical benefits over Carnapian-Goguenism:

- The new, revised definition of **o**ntology is based on syntax presentation and is not restrictive of any kind of semantics, allowing one to attach novel and non-relational semantics to **o**ntologies.
- Fibring-based ontological connection does not force generated **o**ntologies to have certain properties regardless of input **o**ntologies. Furthermore, more complex combination operations historically derived from fibring can be incorporated into da Costian-Tarskianism to ensure ontological connection preserves additional properties.
- Adding a new logic to an extended development graph does not require rebuilding an entire artificial construct, such as the Grothendieck institutions, since extended consequence systems are structurally simple and abstract enough to not require flattening.
- da Costian-Tarskianism supports one extra heterogeneous operation, namely ontological decomposition. However, it is noteworthy that Carnapian-Goguenism could also account for a similar operation using a categorical universal property construct, similarly to what has been proposed in this section.

For those seeking a concise overview, table 6.4 presents a comparative analysis between Carnapian-Goguenism and da Costian-Tarskianism.

	Carnapian-Goguenism	da Costian-Tarskianism
Structural underpinning	Semantics	Syntax
Philosophical underpinning	Descriptive	Normative
Refinement?	✓	✓
Integration?	✓	✓
Connection?	✓	✓
Decomposition?	✗	✓
Implementation maturity	Production-ready	Baby-steps

Table 6.1: Table summarizing key traits of Carnapian-Goguenism and da Costian-Tarskianism.

[3]https://github.com/AlgebraicJulia

Chapter 7

From Logic to Ontology

UP until this point, the previous chapters discussed the history of **o**ntologies, what they are, which ones exist, how to relate them and what is their relationship to logics. However, in order to fully connect Ontology with Logic, it is necessary to discuss the specific classes of logics that are used to construct **o**ntologies.

The objective of this chapter is to provide a concise overview of the characteristics that distinguish a logic as a suitable candidate to describe **o**ntologies. Additionally, this chapter will examine the logics that have been selected in the literature to implement the **o**ntology description languages described in chapter 3, such as OWL, KIF, CLIF, DAML+OIL, and OBOF.

Finally, this chapter discusses the intricacies of the task of encoding ontological knowledge. The process of encoding ontological knowledge relates to but is not covered by the tasks of choosing a logic to construct an **o**ntology nor the task of developing tools to relate **o**ntologies. Thus, there are certain challenges inherent to this separate task. Describing such challenges concludes the effort to construct a bridge between Applied Ontology and Logic.

7.1 Logics for ontologies

BASED on chapters 4 and 5, it has become evident that logics may be used to describe **o**ntologies and there are at least two possible approaches for doing so. However, the following question remains — which logics are, after all, suitable candidates to describe **o**ntologies? In order to provide an answer to this question, it is first necessary to understand what is meant by the term "suitable candidate", as this is concept is contextual. What follows is an attempt to address this question which draws heavily from Hoekstra's purpose-based abstract classification of **o**ntologies. The interested reader may refer to the book "Ontology Representation: Design Patterns and Ontologies that Make Sense" [179] for further details.

In a theoretical setting, it is clear one should only consider logics whose expressive power[1] is sufficient or even exceeds the requirements to encode the concepts and relations

[1]The term "expressive power" loosely and informally refers to the computational complexity class captured by the logic, in the sense of descriptive complexity. For an overview on this matter, the reader can refer to Libkin [230].

to be modelled by an **o**ntology. For instance, if one desires to construct an **o**ntology whose ontological commitments include standard temporal reasoning, the underlying logic must include quantification and relational reasoning, since the ontological level contains concepts such as instant or moment and relations such as "before" or "after".

Indeed, the argument for expressiveness appears to be a sound route for what Hoekstra [179, p. 80] calls, rather confusingly, *formal **o**ntologies*: **o**ntologies which are merely a formal specification of an ontological theory in philosophy, with minimal ontological bias. This statement can be clarified through examples. DOLCE's **QS5** presentation is a formal **o**ntology in the sense of Hoekstra, since it is not concerned with any other descriptive aspect aside from expressivity. In an attempt to reduce some terminological confusion, Hoekstra's formal **o**ntologies will be renamed to *prototypical* **o**ntologies. Therefore, prototypical **o**ntologies impose no restrictions on which logic should be chosen, aside from the inherent expressivity requirement from ontology design. This usually leads one to choose first-order logic to construct prototypical **o**ntologies.

As chapters 3 and 5 have detailed, it is highly desirable for **o**ntologies to be, in fact, concretely implemented via computer programs interacting with automated reasoners. This, in practice, prevents one from freely use any highly expressive logic. For instance, first-order logic is well known to be undecidable [81, 385], hence it cannot be used as the underlying logic to construct decidable **o**ntologies. **O**ntologies subject to this decidability restriction were named *knowledge representation* **o**ntologies by Hoekstra [179, p. 80]: reusable terminological knowledge representations which specify part of the domain theory, as opposed to the entire ontological theory, using representation languages allowing automated reasoning. Again, in an attempt to minimize confusion, knowledge representation **o**ntologies will be renamed to *reasoning* **o**ntologies. Therefore, it is possible to claim that reasoning **o**ntologies may implement prototypical **o**ntologies. This is the case for one of DOLCE's versions, DOLCE-Ultralite and Dolce-zero — it is represented not in first-order logic, but in OWL.

A reasoning **o**ntology does not necessarily depend on a prototypical **o**ntology. This distinction is indeed part of the multi-dimensional classification of **o**ntologies presented by Hoekstra [179, p. 102], which considers an extra type of **o**ntology called *knowledgement management* **o**ntologies. Hoekstra proposes additional types of **o**ntologies, however reasoning **o**ntologies are sufficient to propose a contextual answer the matter of what are "suitable candidate" logics: logics which, for practical reasons, are necessarily decidable and sufficiently expressive. In other words, suitable candidate logics for **o**ntologies balance computational tractability and expressivity.

Note that the representations of reasoning **o**ntologies may not strictly be logics — for instance, OWL as in DOLCE's case. This is because, in practice, reasoning **o**ntologies require an extra layer of abstraction for ease of use, resulting in the actual *representation languages* in the literature, many of which were presented by chapter 3: OWL, CLIF, KIF, OBOF, DAML+OIL, RDF, and others. Therefore, separate to the matter of which logics directly describe **o**ntologies, there lies the question of what is, then, the underlying logic of the *representation languages*?

The answer is, nearly in ubiquity, description logics — the topic of the next section. Some representation languages, as CLIF and KIF, are not formalized by description logics but first-order logic — thus, they are not classified as "suitable candidates" for reasoning **o**ntologies if one considers the previously stated definition. This also underscores the fact that using a concrete representation language, e.g. CLIF or KIF, to describe an **o**ntology is not enough to classify it as a reasoning **o**ntology. The fact that an **o**ntology enjoys a concrete implementation is not sufficient[2] to classify it as a "reasoning" **o**ntology.

7.2 Description Logics

In short, description logics are usually decidable fragments of first-order logic with differing syntax. This assertion is likely to be anticipated by the reader, given the preceding discussion. Nevertheless, the history of description logics is similarly intriguing, paralleling that of applied Ontology. It is situated at the intersection between Logic and Artificial Intelligence. The goal of this brief section is to provide a short historical account of how description logics came to be, along with some examples and some of their interesting properties.

From this point forward, the abbreviation DL will be used to refer to a specific description logic and DLs will be used to refer to a class of description logics. The principal references in this section are the nearly universally acknowledged books [23, 170], which are arguably regarded as the definitive works on description logics, and the excellent survey [341].

7.2.1 A history of description logics

Harmelen et al. [170, p. 138-139] divide the history of DLs into five different phases, where phase 0 is the "pre-history" of description logics and phase 4 is current day. The following is a brief examination of these different phases, with proper emphasis on phase 1 onwards, since historically, DLs did not exist in phase 0.

The first DL, named Kl-One, was initially created by Brachman [51] and later fleshed out by Brachman and Schmolze [54]. As Brachman and Schmolze [54] themselves state, Kl-One was initially developed to answer the needs of Artificial Intelligence knowledge representation, however interestingly enough the authors poignantly decided to expose the philosophy behind Kl-One.

Kl-One, and by extension other DLs, arose as a way to overcome a particular deficiency in previous knowledge representation approaches from the pre-historical phase 0, semantic networks [312] and frame-based systems [257]: lack of formal semantics. Brachman and Schmolze [54, p. 174] succinctly state Kl-One provides a language for expressing an explicit set of beliefs, via a knowledge base, for a rational agent. This language provided by Kl-One is in turn divided into two formalisms, one for assertions and another for descriptions or concepts. It will shortly become clear that these correspond to the roots of how all DLs are formally presented.

Phase 1 begins with Kl-One and includes the development of other representation languages mostly concerned with the implementation of systems, such as K-Rep, Krypton, Back Loom. Harmelen et al. [170, p. 138] note systems using such representation languages employed structural subsumption algorithms, one of the simplest

[2]Note that it is not sufficient nor necessary, for one could describe an **o**ntology directly using propositional calculus, and thus it would be considered a reasoning **o**ntology.

inference algorithms and a basis for many others [330]. These algorithms are computationally tractable for a class of DLs, i.e. polynomial in time as shown by Brachman and Levesque [53], and provide a way to normalize concept descriptions and reduce them. The subsumption problem informally refers to the problem of deciding if a concept description can be reduced into another, hence the namesake of these algorithms.

It was quickly shown that the class of DLs for which structural subsumption is tractable contains rather inexpressive DLs. Indeed, Brachman and Levesque [53] show that adding seemingly inconsequential features to a DL to increase its expressiveness may render the structural subsumption algorithm intractable. Furthermore, it was shown by Schmidt-Schaubß [338] that the subsumption problem for the underlying DL of Kl-One is not only intractable but also undecidable. In face of these issues, P. F. Patel-Schneider et al. [290] presented Classic, a then novel knowledge representation system carefully built to balance expressive power and tractability of the subsumption problem [42, 52].

In the early 1990s, there was a shift in interest from structural subsumption algorithms to broader algorithmic horizons and deeper understanding of the complexity of reasoning. Significant research was carried out to improve the reasoning complexity of DLs via tableau-based algorithms to check the satisfiability of a set of formulas or, in DL jargon, the consistency of a knowledge base. Tableau-based algorithms in the context of DLs differ themselves slightly from the usual method of analytic tableau in proof theory, as presented by Beth [36] and further explored by W. A. Carnielli [76] and Smullyan [364]. The first complete tableau-based algorithm for $\mathcal{ALC}$, a particularly expressive DL, was developed by M. S. G. Smolka [363]. The intuitive idea behind their algorithm is it generates a search tree over a knowledge base by repeatedly applying transformation rules. The transformation rules in turn generate branches and nodes containing formulae — if no rules are applicable to a node in the search tree, it is deemed a leaf. The knowledge base is consistent if no leaves contain contradictions and, additionally, it is *satisfiable*[3]. Kris [19] and Crack were one of the first systems to employ tableau-based algorithms for reasoning resulting in acceptable behavior, despite the fact that such algorithms are not polynomial unlike the — as a matter of fact, the tableau-based algorithm for $\mathcal{ALC}$ was shown to be PSPACE-complete [22].

It is noteworthy that tableau-based methods are frequently employed in the context of modal logics, particularly as a proof procedure [77, 80]. Schild [337] showed $\mathcal{ALC}$ is a notational variant of the multi-modal logic $\mathbf{K}(m)$, initiating third phase of DL history. This phase focused on building highly optimized systems based on tableau-based algorithms, and on establishing relationships between DLs and other classes of logics, particularly modal logics and fragments of first-order logic [23, p. 139]. During this period, a number of highly optimized reasoning systems were developed, including FaCT [184], RACE [168] and DLP [288]. Phase 3 was also marked by a surge in interest towards database applications, such as the work developed by Calvanese, De Giacomo, and Lenzerini [64] on query optimization, Palopoli, Saccà, and Ursino's and Buchheit, Donini, and Schaerf's work on schema reasoning [56, 285].

There was no discernible turning point between phase 3 and phase 4, which is the current phase. The DL literature from the 2000s may be divided into several categories,

[3]This is a key difference from traditional analytic tableaux. Traditionally, one constructs a single formula by negating the conjunction of each element of a set of formulae to determine whether the set is satisfiable. However, for propositionally closed DLs, satisfiability reduces to consistency and thus this extra step is not necessary. See Baader et al. [23, p. 65-74] for details.

all of which build upon the results of previous phases. A few of these categories are: works focused on specific applied domains [91, 92, 177, 217], theoretical work exploring different kinds of DLs (such as substructural, fuzzy and temporal DLs) [12, 21, 373], implementations of DLs via representation languages for general and specific domains [141, 142, 192], and neuro-symbolic reasoning [259, 400].

As a matter of fact, phase 3 reasoning systems formed the basis for those now currently in use practice, such as *fuzzy*DL [41], Pronto [207], Pellet [355], FaCT++[4] [384]. In general, a significant amount of effort has been invested in the standardization and enhancement of the robustness of DL-based systems. For instance, the World Wide Web Consortium (W3C) accepted DLs as the underlying formalism powering the Semantic Web, hence why OWL [332] and OWL [178] are based on DLs, a fact foreshadowed in chapter 3.

In summary, modern DLs are defined by Baader et al. [23] as

> Fully fledged logics with formal semantics, usually decidable fragments of first-order logic (FOL) closely related to modal, dynamic logics and the guarded fragment[5].

DLs have very well-understood computational properties, such as decidability, complexity of satisfiability and theory consistency checking, finite model and tree property validity. Additionally, DLs enjoy highly optimized systems with practical decision procedures for key problems, such as satisfiability, subsumption, equivalence and query answering.

7.2.2 A family of description logics

Having established the historical context and essential characteristics of DLs, it is now possible to define and examine one such DL. In fact, this section examines a family of DLs derived from $\mathcal{ALC}$, the Attributive Concept Language with Complements, by inspecting how it may be extended by adding or removing certain axioms and properties.

As Baader and Lutz [20] note, $\mathcal{ALC}$ is the smallest propositionally closed DL, meaning its logical connectives are functionally complete, i.e. all boolean set operations can be expressed. It was introduced by Schmidt-Schauß and G. Smolka [339] with the explicit objective of reducing the aforementioned subsumption problem into the satisfiability problem, an objective which was successfully achieved. $\mathcal{AL}$, i.e. the Attributive Concept Logic, forms the basis of a family of DLs, of which $\mathcal{ALC}$ is a part. However it is not the only "family-generator" DL, other examples being the sub-boolean logics $\mathcal{EL}$, Existential Logic, and $\mathcal{FL}$, the Frame-Based Description Language [228]. However, the reason to focus on $\mathcal{ALC}$ is that it is tied to the overall goal of this chapter, i.e. espouse the underlying logics of the **o**ntologies described in chapter 3.

The following definitions are due to Harmelen, Lifschitz, and Porter [169], the first of which is that of a signature of a DL.

[4] As a remark, FaCT++ is supported by the Hets program described in chapter 5.

[5] The guarded fragment, as defined by Andréka, Németi, and Benthem [8], is an extension of modal logic into a richer fragment of first-order logic with bounded quantifiers. It enjoys several meta-properties, among which decidability is one of them.

Definition 7.2.1 (Signature of a DL)**.** The *signature* of a DL is a triple $\langle N_C, N_R, N_I \rangle$, where:

- N_C is the set of *concept names* (e.g. Cat, $Husband$, $Wife$, $\ldots$), equivalent to FOL unary predicates,
- N_R is the set of *role names* (e.g. *sits-on*, *loves*, $\ldots$), equivalent to FOL binary predicates,
- N_I is the set of *individual names* (e.g. $Felix$, $John$, $Mary$), equivalent to FOL constants.

The signature of a DL is not specific to $\mathcal{ALC}$, meaning other DLs also leverage the same kind of signature. Additionally, the signature does *not* include the logical connectives. This is because DLs are implicitly or intuitively thought of as "extending" or somehow "relying" on FOL, thus the signature represents the extra-logical symbols for knowledge representation and **o**ntology building purposes. Recalling chapter 7, the signature of a DL can be intuitively understood as the signature of the *ontological aspect*.

The following defined the syntax of $\mathcal{ALC}$.

Definition 7.2.2 ($\mathcal{ALC}$ syntax)**.** Given a *signature* $\langle N_C, N_R, N_I \rangle$, the set of all *$\mathcal{ALC}$-concept descriptions* or simply *$\mathcal{ALC}$-concepts*, is the smallest set such that:

1. $\top$, $\bot$ and every $A \in N_C$ is an $\mathcal{ALC}$-concept
2. if C and D are $\mathcal{ALC}$-concepts and $r \in N_R$, then $C \sqcap D$, $C \sqcup D$, $\neg C$, $\forall r.C$ and $\exists r.C$ are $\mathcal{ALC}$-concepts.

$\mathcal{ALC}$-concepts are "syntatic sugar"[6] for FOL formulae. For instance, $Cat \sqcap \exists sits\_on.Mat$ is equivalent to $\exists y\ (Cat(x) \wedge sits\_on(x, y))$. $\top$ and $\bot$ may be thought of as "thing" and "nothing" concepts, respectively. They are also not necessarily primitive as one could define $\top =_{def} A \sqcup \neg A$ and $\bot =_{def} A \sqcap \neg A$.

Definition 7.2.3 ($\mathcal{ALC}$ semantics)**.** An interpretation $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ consists of a non-empty set $\Delta^{\mathcal{I}}$ (the domain of $\mathcal{I}$) and a map $\cdot^{\mathcal{I}}$ which takes every $\mathcal{ALC}$-concept to a subset of $\Delta^{\mathcal{I}}$ and every role name to a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ such that for all concepts C, D and role names r:

$$\top^{\mathcal{I}} = \Delta^{\mathcal{I}} \quad \bot^{\mathcal{I}} = \Delta^{\mathcal{I}}$$

$$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}} \quad (C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}} \quad (\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$$

$$(\exists r.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} : \text{There is some } y \in \Delta^{\mathcal{I}} \text{ with } \langle x, y \rangle \in r^{\mathcal{I}}, y \in C^{\mathcal{I}}\}$$

$$(\forall r.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} : \text{For all } y \in \Delta^{\mathcal{I}},\ \text{if } \langle x, y \rangle \in r^{I},\ \text{then } y \in C^{\mathcal{I}}\}$$

It is note-worthy that set-theoretical semantics for $\mathcal{ALC}$ constitute an interpretation system in the sense of W. Carnielli et al. [74].

Given a DL, it is desirable to formalize knowledge based on the concepts, roles and individuals of a particular domain. This is done by introducing and assuming two kinds of axioms:

[6]This term was defined by Landin [215] in the context of programming languages to define syntax which is not strictly necessary and may be substituted by more fundamental syntactical constructs.

- "schema" or *terminological* axioms - corresponding to particular forms of FOL sentences
- "data" or *assertional* axioms - corresponding to FOL formulae with no variables

In the jargon of DLs, a terminological axiom is also called general concept inclusion, and it is defined as follows.

Definition 7.2.4 (General Concept Inclusion)**.** A *general concept inclusion* (GCI) is of the form $C \sqsubseteq D$, where C, D are $\mathcal{ALC}$-concepts. An interpretation $\mathcal{I}$ is a model of a GCI $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, we may write $\mathcal{I} \models C \sqsubseteq D$. $C \equiv D$ is an abbreviation for the pair of GCIs $C \sqsubseteq D$ and $D \sqsubseteq C$.

A GCI is the most general form of a terminological axiom, as it corresponds to FOL implication. A set of terminological axioms is called a terminological box or TBox.

Definition 7.2.5 (Terminological Box)**.** A finite set of GCIs is called a *TBox* (terminological box). An interpretation $\mathcal{I}$ is a model of a TBox $\mathcal{T}$ if it is a model of every GCI in $\mathcal{T}$.

Assertional axioms are defined in the following manner.

Definition 7.2.6 (Assertional Axiom)**.** An *assertional axiom* is of the form $x : C$ or $(x, y) : r$ where C is an $\mathcal{ALC}$-concept, r is a role name and x and y are individual names. An interpretation $\mathcal{I}$ is a model of an assertional axiom $x : C$ if $x^{\mathcal{I}} \in C^{\mathcal{I}}$ and a model of an axiom $(x, y) : r$ if $\langle x^{\mathcal{I}}, y^{\mathcal{I}} \rangle \in r^{\mathcal{I}}$.

Similarly to a TBox, an assertional box or ABox is a collection of assertional axioms.

Definition 7.2.7 (Assertional Box)**.** A finite set of assertional axioms is called an *ABox*. An interpretation $\mathcal{I}$ is a model of an ABox $\mathcal{A}$ if it is a model of every axiom in $\mathcal{A}$.

Finally, a knowledge base is defined as an ABox paired with a TBox.

Definition 7.2.8 (Knowledge Base)**.** A *knowledge base* (KB) is a pair $(\mathcal{T}, \mathcal{A})$, where $\mathcal{T}$ is an TBox and $\mathcal{A}$ is an ABox. An interpretation $\mathcal{I}$ is a model of a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ if it is a model of $\mathcal{T}$ and $\mathcal{A}$.

Intuitively, a knowledge base can be thought of, quite simply, as a collection of particular FOL axioms. However, this concept is crucial in that it relates to the *ontological aspect*, representing the ontological axioms.

Example 7.2.1. Consider the signature

$$\langle \{Person, Parent, HappyParent\}, \{hasChild\}, \{John, Mary\} \rangle$$

For this signature, we may have the following TBox:

$$\begin{aligned} \mathcal{T} = \{ Doctor &\sqsubseteq Person, \\ Parent &\equiv Person \sqcap \exists hasChild.Person, \\ HappyParent &\equiv Parent \sqcap \forall hasChild.(Doctor \sqcup \exists hasChild.Doctor) \} \end{aligned}$$

and the following ABox:

$$\mathcal{A} = \{John : HappyParent,\\ (John, Mary) : hasChild\}$$

Then a knowledge base $\mathcal{K}$ is simply the pair $(\mathcal{T}, \mathcal{A})$.

Getting into the extensions of $\mathcal{ALC}$, they are broadly classified as "classical" and "non-classical" extensions, where "classical" means the semantics of the extended logic can be described using the set-theoretical semantics from definition 7.2.3. As previously mentioned, $\mathcal{ALC}$ is *itself* an extension of the weaker, negation-less DL $\mathcal{AL}$ — the negation operator is an extension denoted by the letter $\mathcal{C}$. A few "classical" extensions of $\mathcal{AL}$ will be presented to illustrate this concept. The terms ABox, TBox and knowledge base are also applicable in the context of other DLs based on $\mathcal{ALC}$.

As noted by Baader [17, p. 496], there are three main ways to classically extend $\mathcal{ALC}$: via restrictions on role semantics, role constructors and concept constructors. Restrictions on role semantics take a subset of the overall set N_R of role names in order to generate new logics, two example restrictions are:

- Functional roles: take the subset $N_F \subseteq N_R$ such that every role $f \in N_F$, called a *feature*, corresponds to a functional binary relation $f^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ via an interpretation $\mathcal{I}$. Extending $\mathcal{AL}$ with functional roles, one obtains $\mathcal{AL}_f$.
- Transitive roles: take the subset $N_{R^+} \subseteq N_R$ such that every role $R \in N_{R^+}$ corresponds to a transitive binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. $\mathcal{AL}$ extended with transitive roles is denoted by $\mathcal{AL}_{\mathcal{R}^+}$ and, for brevity in certain contexts, by $\mathcal{S}$.

Concept constructors generate new concept descriptions from existing concept and role descriptions and establish semantics for the new syntax. One such constructor is $\mathcal{C}$, whose syntax is $\neg C$ and semantics is the set-theoretic negation. Table 7.2.2 partially reproduces the table of concept constructors from [17, p. 497].

Name	Syntax	Semantics	Symbol
Negation	$\neg C$	$\Delta^{\mathcal{I}}$	$\mathcal{C}$
Qualified	$\geq_n R.C$	$\{a \in \Delta^{\mathcal{I}} \mid \lvert\{b \in \Delta^{\mathcal{I}}\}\rvert \geq n\}$	
number	$\leq_n R.C$	$\{a \in \Delta^{\mathcal{I}} \mid \lvert\{b \in \Delta^{\mathcal{I}}\}\rvert \leq n\}$	$\mathcal{Q}$
restriction	$=_n R.c$	$\{a \in \Delta^{\mathcal{I}} \mid \lvert\{b \in \Delta^{\mathcal{I}}\}\rvert = n\}$	
Nominal	I	$I^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ with $\lvert I^{\mathcal{I}}\rvert = 1$	$\mathcal{O}$
Agreement and	$u_1 \doteq u_2$	$\{a \in \Delta^{\mathcal{I}} \mid \exists b \in \Delta^{\mathcal{I}}.u_1^{\mathcal{I}}(a) = b = u_2^{\mathcal{I}}(a)\}$	$\mathcal{F}$
disagreement	$u_1 \doteq u_2$	$\{a \in \Delta^{\mathcal{I}} \mid \exists b_1, b_2 \in \Delta^{\mathcal{I}}.u_1^{\mathcal{I}}(a) = b_1 \neq b_2 = u_2^{\mathcal{I}}(a)\}$	

Table 7.1: Table with example concept constructors of description logics.

Similarly, role constructors build more complex role descriptions based on existing roles or concept descriptions. Table 7.2.2 presents example role constructors, also partially reproducing and adapting a similar table from [17, p. 499].

DLs may, thus, be referred to by their base DL and its extensions. For instance, the logic $\mathcal{ALC}_{\mathcal{R}^+}\mathcal{QOF}$ denotes the logic $\mathcal{ALC}$ plus role transitivity, qualified number

Name	Syntax	Semantics	Symbol
Inverse	R^-	$\{(b,a) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}}\}$	$\mathcal{I}$ or -1
Composition	$R \circ S$	$R^{\mathcal{I}} \circ S^{\mathcal{I}}$	$\circ$
Role hierarchy	$R \sqcap S$	$R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$	$\mathcal{H}$
Complex role inclusions	$R \circ S \sqcap R$ $R \circ S \sqcap S$	$R^{\mathcal{I}} \circ R^{\mathcal{I}} \subseteq R^{\mathcal{I}}$ $R^{\mathcal{I}} \circ R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$	$\mathcal{R}$

Table 7.2: Table with example role constructors of description logics.

restrictions, nominals and agreement and disagreement. Since referring to DLs by their properties can quickly become a mouthful, it is not unusual for DLs to have abbreviations. For instance, the logic $\mathcal{ALC}_{R+}$ is at times referred by the abbreviation $\mathcal{S}$.

One particular DL stands out in the context of applied Ontology. $\mathcal{SROIQ}$, i.e. $\mathcal{ALC}$ with complex role inclusion, nominals, inverse roles and qualified number restrictions, is the underlying logic of the sub-language OWL2 of OWL [183]. $\mathcal{SROIQ}$ was proposed by Horrocks, Kutz, and Sattler [185] as an improvement over the DLs $\mathcal{SHOIN}$ and $\mathcal{SHOIQ}$. Horrocks, Kutz, and Sattler proposed $\mathcal{SROIQ}$ an answer to the matter of defining a DL whose key inference problems, such as subsumption and satisfiability, are decidable and sufficiently tractable. The table 7.2.2 below maps existing representation languages to their underlying description logics. It may be noted, however, that not all representation languages are based on DLs, such as CLIF.

Representation Language	Description logic
OIL	$\mathcal{SHIQ}$ [120]
OWL-DL	$\mathcal{SHOIN}^{(\mathcal{D})}$ [186]
OWL Lite	$\mathcal{SHOIN}^{(\mathcal{D})}$ [186]
OWL2	$\mathcal{SROIQ}$ [185]

Table 7.3: Table with representation languages and their underlying description logics.

Current research trends in the field of description logics include extending existing description logics with concept and role constructors, with the objective of improving reasoning complexity and ease of use. For instance, recent works developed by Baader and Gil [18] and Jackermeier, Chen, and Horrocks [193] extend the DL $\mathcal{EL}$ with additional operators. Another trend in the literature involves extending DLs to incorporate non-classical properties. This is exemplified by the works of Dalmonte et al. [107] and Artale et al. [13] on non-normal modal description logics, which aim to increase expressiveness. Furthermore, there is ongoing work to extend description logics with non-classical properties, such as the Bayesian description logic $\mathcal{BALC}$ [47].

7.3 Encoding ontological knowledge

There are challenges inherent to encoding ontological knowledge in real-world applications, termed Ontology-based Data Access (OBDA) systems by T. Schneider and Šimkus [341]. OBDA systems are computer systems constructed upon **o**ntologies, with the objective of facilitating heterogeneous data integration. They typically connect one or more data sources (e.g., databases, data lakes, data warehouses, structured data or simply text files) for the purposes of reasoning and querying. This section examines some challenges in the development of OBDA systems, with a particular focus on finite reasoning, open-world vs. closed-world assumptions, and inconsistency-tolerant query answering. The reader may refer to the excellent surveys [39, 341] for comprehensive accounts of ontological knowledge management matters.

Trakhtenbrot [382] proved the widely-known result that finite model satisfiability is undecidable in first-order logic. As a consequence of this, most of the works on reasoning in DLs assume that the domain of models is infinite for tractability reasons[7]. However, this assumption is at times incompatible with the domain one wants to model. As T. Schneider and Šimkus [341] note, an example domain where the models must be finite is that of relational databases, and, in general, the same holds for any domain of computational objects.

Notwithstanding the fact that reasoning algorithms over DLs frequently assume infinite models in the domain, T. Schneider and Šimkus [341] note that OBDA systems often make the *open-world assumption*. The open-world assumption, as opposed to the closed-world assumption, loosely posits that if a fact is not present in an OBDA system, then such a fact is *not* assumed to be false. To be more precise, if a fact is not in the deductive closure of an OBDA system's set of facts (corresponding to, for instance, information in one or more databases), it is not false. This is in contrast to systems that make the *closed-world assumption*, that is, if a fact is not present in the system's set of facts, it is assumed to be false.

The *open-world assumption* has several consequences for OBDA systems. Primarily, it is a strong assumption that does not permit one to consider certain parts of the system's fact base as complete or incomplete. That is, it forces the system to consider the entirety of its fact base as incomplete. In order to overcome this limitation, T. Schneider and Šimkus [341] note the following lines of research: the theory of closed predicates [122], which impacts the tractability of reasoning [277]; and the development of non-monotonic DLs, DLs that allow inference of new facts from the absence of information [55]. In general, the issue of integrating the closed-world and open-world assumptions remains unresolved and is still attracting considerable interest.

Another challenge pertaining to OBDA systems is the development and implementation of *inconsistency-tolerant reasoning*. In contrast to incomplete reasoning, as introduced by the open-world assumption, inconsistency-tolerant reasoning refers to a scenario in which two facts in an OBDA system contradict each other. The Principle of Explosion, discussed in chapter 5, states that a contradiction leads to triviality in systems adhering to classical semantics[8]. Consequently, it is highly desirable for such systems to not contain contradictions.

However, Bienvenu [39] and T. Schneider and Šimkus [341] both note that, in real-world data, quality problems and inconsistent information are nearly unavoidable. To address this, numerous authors have proposed modifying the semantics of DLs to allow for reasoning over inconsistent knowledge bases. Bienvenu [39] presents nine different semantics which have been presented and examined in the literature, highlighting that the fundamental concept underlying most of such semantics is that of ABox repairs. This

[7] Finite model satisfiability was in fact proven to be decidable in several DLs by Calvanese [62].

[8] Provided that the system uses an underlying logic that is complete, as inconsistency is a syntactical concept.

concept is due to Lembo et al. [224] and it is formalized as follows.

Definition 7.3.1 (ABox Repair)**.** An *(ABox) repair* of an ABox $\mathcal{A}$ with respect to a TBox $\mathcal{T}$ is an inclusion-maximal subset of A that is T-consistent. $Rep(\mathcal{A}, \mathcal{T})$ denotes the set of repairs of $\mathcal{A}$ with respect to $\mathcal{T}$, which may be abbreviated to $Rep(\mathcal{K})$ when $K = \langle \mathcal{A}, \mathcal{T} \rangle$.

As Bienvenu [39, p. 2] writes, an ABox repair is, intuitively, a way to restore consistency while retaining as much information as possible. The set of repairs of an ABox represents, then, all the different ways of restoring consistency. In general, inconsistency-tolerant semantics re-define satisfiability (or semantic entailment) in terms of ABox repairs.

Extending the reasoning capabilities OBDA systems based on DLs via inconsistency-tolerant semantics leads to further challenges. A few examples of such challenges are: a rise in reasoning complexity; the issue of explainability of reasoning; and the issue of constructing ABox repairs. Bienvenu [39, p. 7] again details the complexity of query answering given each inconsistency-tolerant semantics they examined for the DLs $\mathcal{ALC}$ and $\mathcal{EL}_{\vdash}$. Explainability, in this context, refers to the ability of a system to explain its own reasoning when providing an answer to a query, e.g., whether a fact follows from the set of facts in an OBDA system. The issues of explainability and constructing ABox repairs are related, since a system with inconsistency-tolerant semantics may provide an answer to a user's query by repairing an ABox in a particular manner that does not coincide with the user's intentions. Explainability is a topic of great depth with its own challenges, therefore it is not in scope for this chapter to discuss it at length. However, it is noteworthy that explainability is not only a concern in ontological systems, but also in the field of Artificial Intelligence as a whole. The reader may refer to the works of Baclawski et al. [25], Bourguin et al. [49], Marques-Silva [248] and Alrabbaa et al. [7] on the broader subject of explainability.

It should be noted that inconsistency-tolerant semantics does not permit reasoning with inconsistencies. As its name suggests, such semantics tolerates inconsistencies by, in a somewhat informal sense, considering scenarios where the inconsistencies do not exist or have been resolved. Moreover, these semantics do not permit the inference of facts from inconsistencies. This is because the underlying assumption in proposing inconsistency-tolerant semantics is that inconsistencies should be acknowledged but avoided. However, in certain domains of knowledge, inconsistent sets of information do not trivialize in practice. The law is one such domain, as evidenced by the works of Engel [118] and Duck-Mayr [115].

In light of the inherent characteristics of inconsistency-tolerant semantics, it may be beneficial to reconsider the assumption that inconsistencies should be avoided. By reconsidering the assumption and acknowledging that inconsistencies may represent useful information, it is necessary to develop DLs whose syntax and reasoning rules allow for inference even in the presence of inconsistencies. This corresponds to developing paraconsistent DLs in the sense of N. C. A. d. Costa [98], that is, non-trivial logics for which the principle of explosion does not hold. It is therefore appropriate to present a few examples of such DLs.

It can be argued that first paraconsistent DL is Patel-Schneider's inconsistency-tolerant four-valued terminological logic [289]. However, Patel-Schneider did not present his logic as part of the paraconsistent enterprise; rather, his main objective was to improve the tractability of the subsumption problem. On the other hand, S. P. Odintsov and Wansing [280] and Ma, Hitzler, and Lin [236] have proposed paraconsistent DLs with the explicit objective of allowing reasoning with inconsistencies [236, 237, 279, 280]. One such DL is $\mathcal{ALC}4$ [236], an extension of $\mathcal{ALC}$ with decidable four-valued semantics analogous to Belnap and Dunn's useful four-valued logic or first degree entailment [33, 34, 117]. In his paper, Kamide [202] presents an overview of other paraconsistent DLs proposed in the literature, along with $\mathcal{PALC}$, a paraconsistent extension of $\mathcal{ALC}$ analogous to Nelson's paraconsistent four-valued logic with strong negation N4 [272].

The novel paraconsistent DLs being proposed by W. Carnielli, M. E. Coniglio, and Bueno-Soler [73] in a forthcoming paper, based on previous work by Carnielli and Bueno-Soler, may be regarded as suitable candidates for not only allowing inconsistency-tolerant reasoning, but also integrating open-world and closed-world assumptions. W. Carnielli, M. E. Coniglio, and Bueno-Soler [73] propose a family of DLs which extend $\mathcal{ALC}$ by inherting traits of the logics **RmbC** and **RmbCciw**, which are Logics of Formal Inconsistency (LFIs) with replacement [72, 75]. The LFIs can be informally thought of as paraconsistent logics which are able to internalize the notion of consistency (or inconsistency) via an operator $\circ$, a trait that distinguishes them from other paraconsistent logics.

W. Carnielli, M. E. Coniglio, and Bueno-Soler [73] tentatively name their paraconsistent DL $\mathcal{PALC}$, however to distinguish it from Kamide's logic, their logic shall be called $\mathcal{ALC}_{\mathcal{LR}}$, where $\mathcal{L}$ stands for LFIs and $\mathcal{R}$ stands for 'replacement'. The following is a brief examination of $\mathcal{ALC}_{\mathcal{LR}}$'s syntax and semantics, as per W. Carnielli, M. E. Coniglio, and Bueno-Soler [73]. It is necessary to note that, similarly to $\mathcal{ALC}$, several logics may be generated by specifying different signatures. Thus, at times, $\mathcal{ALC}_{\mathcal{LR}}$ is referred to in plurality to denote this fact.

Definition 7.3.2 (Syntax of $\mathcal{ALC}_{\mathcal{LR}}$)**.** The syntax of an $\mathcal{ALC}_{\mathcal{LR}}$ extends the syntax of $\mathcal{ALC}$ by:

- Denoting classical negation by $\sim$;
- Adding the paraconsistent negation operator $\neg$;
- Adding the consistency operator $\circ$.

Let Ω be the first-order signature underlying an $\mathcal{ALC}_{\mathcal{LR}}$ and $\Sigma = \{\wedge, \vee, \rightarrow, \neg, \circ\}$. The first-order language over Ω and Σ is denoted by $For_1(\Omega)$.

Syntactically, $\mathcal{ALC}_{\mathcal{LR}}$ does not diverge significantly from $\mathcal{ALC}$. As previously stated, the consistency operator is what permits $\mathcal{ALC}_{\mathcal{LR}}$ to internalize the concept of consistency.

Definition 7.3.3 (Neighborhood first order-structure for $\mathcal{ALC}_{\mathcal{LR}}$)**.** Let $\mathcal{F}$ be a given $\mathcal{ALC}_{\mathcal{LR}}$ over a signature Ω and let U and W be non-empty sets. A *neighborhood first-order structure* for $\mathcal{F}$ is a structure:

$$\mathcal{I} = \langle U, W, S_\neg, S_\circ, \cdot^\mathcal{I} \rangle$$

such that $S_\neg : \mathcal{P}(W) \rightarrow \mathcal{P}(W)$, $S_\circ : \mathcal{P}(W) \rightarrow \mathcal{P}(W)$ are functions and $\cdot^\mathcal{I}$ is an interpretation map over the symbols of Ω defined as follows:

- If a is a constant symbol, then $a^{\mathcal{I}}$ is an element of U;
- If A is an atomic concept, then $A^{\mathcal{I}} : U \to \mathcal{P}(W)$ is a function;
- If R is an atomic role, then $R^{\mathcal{I}} : U \times U \to \mathcal{P}(W)$ is a function.

Definition 7.3.4 (Semantics of $\mathcal{ALC}_{\mathcal{LR}}$)**.** Let $\mathcal{I}$ be a neighborhood first-order structure for an $\mathcal{ALC}_{\mathcal{LR}}$ $\mathcal{F}$. The interpretation map $\cdot^{\mathcal{I}}$ defined for $\mathcal{ALC}$ may be extended to the set of constructors $For_1(\Omega)$ as a map $\cdot^{\mathcal{I}} : For_1(\Omega) \to (\mathcal{P}(W)^U \cup \mathcal{P}(W)^{U\times U})$ recursively defined as follows, for concepts C and D, role name R and $x \in U$:

$$\begin{aligned}
(C^{\mathcal{I}} \sqcap D^{\mathcal{I}})(x) &= C^{\mathcal{I}} \cap D^{\mathcal{I}}(x) && (\textit{conjunction})\\
(C^{\mathcal{I}} \sqcup D^{\mathcal{I}})(x) &= C^{\mathcal{I}} \cup D^{\mathcal{I}}(x) && (\textit{disjunction})\\
(\sim C^{\mathcal{I}})(x) &= W \setminus C^{\mathcal{I}}(x) && (\textit{classical negation})\\
(\neg C^{\mathcal{I}})(x) &= (W \setminus C^{\mathcal{I}}(x)) \cup S_{\neg}(C^{\mathcal{I}}(x)) && (\textit{paraconsistent negation})\\
(\circ C^{\mathcal{I}})(x) &= (W \setminus (C^{\mathcal{I}}(x) \cap (\neg C)^{\mathcal{I}}(x))) \cap S_{\circ}(C^{\mathcal{I}}(x)) && (\textit{consistency})\\
(\forall R.C)^{\mathcal{I}}(x) &= \bigcap_{y\in U}(R^{\mathcal{I}}(x,y) \to C^{\mathcal{I}}(y))\\
&= \bigcap_{y\in U}((W \setminus R^{\mathcal{I}}(x,y)) \cup C^{\mathcal{I}}(y)) && (\textit{value restriction})\\
(\exists R.C)^{\mathcal{I}}(x) &= \bigcup_{y\in U}(R^{\mathcal{I}}(x,y) \cap C^{\mathcal{I}}(y)) && (\textit{exists restriction})\\
(R^{-})^{\mathcal{I}}(x,y) &= R^{\mathcal{I}}(y,x) && (\textit{inverse role})\\
(R^{*})^{\mathcal{I}}(x,y) &= \bigcup_{n\geq 0}(R^{*}_{n})^{\mathcal{I}},\ \textit{where}\\
(R^{*}_{0})^{\mathcal{I}}(x,y) &= R^{\mathcal{I}}(x,y)\ \textit{and}\\
(R^{*}_{n+1})\mathcal{I}(x,y) &= (\bigcup_{z\in U}((R^{*}_{n})^{\mathcal{I}}(x,z) \cap (R^{*}_{n})^{\mathcal{I}}(z,y))) \cup (R^{*}_{n})\mathcal{I}(x,y) && (\textit{transitive role})
\end{aligned}$$

Satisfiability of TBoxes and ABoxes of $\mathcal{ALC}_{\mathcal{LR}}$ is then defined in terms of the map $\cdot^{\mathcal{I}}$ of a neighborhood first-order structure.

Definition 7.3.5 (Satisfiability in $\mathcal{ALC}_{\mathcal{LR}}$)**.** Let $\mathcal{I}$ be a neighborhood first-order structure for an $\mathcal{ALC}_{\mathcal{LR}}$ $\mathcal{F}$.

$\mathcal{I}$ satisfies an assertion of the TBox $\mathcal{T}$ of $\mathcal{F}$ in the following cases:

1. $A \equiv C$ *(concept definition)*: then, $A^{\mathcal{I}}(x) = C^{\mathcal{I}}(x)$ for every $x \in U$;
2. $C_1 \sqsubseteq C_2$ *(general concept inclusion)*: then, $(C_1)^{\mathcal{I}}(x) \subseteq (C_2)^{\mathcal{I}}(x)$ for every $x \in U$.

$\mathcal{I}$ is said to satisfy T if it satisfies all assertions in T.

Similarly, I satisfies an assertion of the *ABox* $\mathcal{A}$ of $\mathcal{F}$ in the following cases:

1. $a : C$ *(concept assertion)*: then, $C^{\mathcal{I}}(a^{\mathcal{I}}) = W$;
2. $\langle (a_1, a_2) \rangle : R$ *(role assertion)*: then, $R^{\mathcal{I}}((a_1)^{\mathcal{I}}, (a_2)^{\mathcal{I}}) = W$.

$\mathcal{I}$ satisfies $\mathcal{A}$ if it satisfies every assertion of A.

The syntax and semantics of $\mathcal{ALC}_{\mathcal{LR}}$ allow one to, in a manner of speaking, mark certain assertions as being "hard" or "soft" in terms of reasoning. That is, the assertion $\circ C$ informally states that the concept C does not admit contradictions, thus if assertions C and $\neg C$ were to be both present in a knowledge base, this would constitute an undesirable inconsistency likely requiring manual intervention. Conversely, if the assertion $\circ C$ is *not* present in a knowledge base, it is admissible to assert that both C and $\neg C$ may be satisfiable under some interpretation.

The general idea of marking certain assertions as consistent may be thought of as implicitly dividing a knowledge base. Assertions marked as consistent may be subject to classical reasoning methods, whereas those not marked may be subject to paraconsistent reasoning methods. This induced split in the knowledge base is not enough to provide a characterization of, or means of, integrating open-world and closed-world assumptions. Nevertheless, the paraconsistent semantics $\mathcal{ALC}_{\mathcal{LR}}$ may be utilized to assess what information may be inferred from a set of assertions based on whether they are marked as consistent or not. Furthermore, paraconsistent semantics may serve as the foundations of a broader effort to characterize open-world and closed-world integration.

Further details on $\mathcal{ALC}_{\mathcal{LR}}$ may be found in [73]. As noted by T. Schneider and Šimkus [341], there exist several challenges in ontological knowledge management aside from those mentioned here (e.g. so-called Horn DLs, the problem of efficient query-rewriting, evolving data management). Finally, it may be argued that every challenge in the development of OBDA systems can potentially lead to the development of extensions of DLs, algorithms over DLs, or the creation of new DLs altogether. Consequently, the fields of applied Ontology and data management are experiencing a period of significant growth, with a plethora of new approaches and proposals to address the aforementioned challenges.

7.4 Description logics as extended consequence systems

BEFORE closing this chapter, this section presents a consolidated and summarized approach to represent **o**ntologies described by representation languages based on DLs, such as those discussed in chapter 2, as extended consequence systems. By representing, for instance, BFO as an extended consequence system, one may utilize the da Costian-Tarskian approach from chapter 6 to achieve ontological interoperability. It is not the goal of this section to provide a detailed account of how to automatically map existing DL-based **o**ntologies into extended consequence systems and vice-versa. However, this section further illustrates the feasibility of such an automated mapping.

It may be of benefit to initiate the discussion by considering a tangible, so-called "toy" **o**ntology, with fewer classes and axioms compared to top-level **o**ntologies. The "course **o**ntology" in example 7.4.1, built exclusively for demonstration purposes, will serve as a recurring example for the time being. Note that the way the **o**ntology is presented does not match any past representation language and it is closer to the syntax of DLs, with slightly different terminology. The appendix A presents the same **o**ntology in the RDF representation language.

Example 7.4.1 (Course **o**ntology)**.** The course **o**ntology O_{ex} has the following structure.

Classes

- Course
- Person
- Professor
  - Professor $\sqsubseteq$ Person
- Student
  - Student $\sqsubseteq$ Person

Object properties

- **takes**
  - **takes** $\sqsubseteq$ **topObjectProperty**
  - AsymmetricProperty(**takes**)
  - IrreflexiveProperty(**takes**)
- **teaches**
  - **teaches** $\sqsubseteq$ **topObjectProperty**
  - AsymmetricProperty(**teaches**)
  - IrreflexiveProperty(**teaches**)
- **taughtBy**
  - **taughtBy** $\sqsubseteq$ **topObjectProperty**
  - **taughtBy** $\equiv$ **teaches**$^{-}$
- **topObjectProperty**

Individuals

- CourseA
  - CourseA : Course
- ProfessorA
  - ProfessorA : Professor
  - {ProfessorA} $\not\equiv$ {ProfessorB}
  - **teaches**(ProfessorA, CourseA)

- ProfessorB
  - ProfessorB : Student
  - ProfessorB : Professor
  - {ProfessorA} $\not\equiv$ {ProfessorB}
  - {ProfessorB} $\not\equiv$ {StudentA}
  - **takes**(ProfessorB, CourseA)
- StudentA
  - StudentA : Student
  - {ProfessorB} $\not\equiv$ {StudentA}
  - **takes**(StudentA, CourseA)

In DL terminology, the **o**ntology O_{ex} has 5 concept names (2 of which are omitted), 3 role names and 4 individual names. Note that the role **taughtBy** is the inverse of **teaches**, thus $\mathcal{ALC}$ is not enough to formally represent O_{ex} as, for instance, the inverse role constructor $\mathcal{I}$ is needed.

However, extended consequence systems are not constrained by a logic one needs to choose. By representing O_{ex} as a consequence system, one does not need to reason about what underlying logic should be used. Nevertheless, it may be beneficial to identify said logic for practical reasons. The example 7.4.2 below represents an attempt to naively represent O_{ex} as an extended consequence system.

Example 7.4.2 (O_{ex} naively represented as an extended consequence system.)**.** O_{ex} may be defined as an extended consequence system $E_{ex} = \langle C, \mathsf{C}, C_o, \Gamma_o \rangle$ such that:

- C_o is a signature such that:
  - $C_0 = \{\top, \bot, ProfessorA, ProfessorB, StudentA, CourseA\}$
  - $C_1 = \{\cdot^-, Course, Person, Professor, AsymmetricProperty, IrreflexiveProperty\}$
  - $C_2 = \{\textbf{takes}, \textbf{teaches}, \textbf{taughtBy}, \textbf{topObjectProperty}, \sqsubseteq, \equiv, \not\equiv\}$ $C_k = \emptyset$ for $k > 2$.
- $C = C_o$ is a signature
- Γ_o is a set containing the following axioms:

1. $Professor \sqsubseteq Person$
2. $Student \sqsubseteq Person$
3. $Person \sqsubseteq \top$
4. $Course \sqsubseteq \top$
5. **takes** $\sqsubseteq$ **topObjectProperty**
6. AsymmetricProperty(**takes**)
7. IrreflexiveProperty(**takes**)
8. **teaches** $\sqsubseteq$ **topObjectProperty**
9. AsymmetricProperty(**teaches**)
10. IrreflexiveProperty(**teaches**)
11. **taughtBy** $\sqsubseteq$ **topObjectProperty**
12. **taughtBy** $\equiv$ **teaches**$^-$
13. $Course(CourseA)$
14. $Professor(ProfessorA)$

15. $ProfessorA \not\equiv ProfessorB$
16. **teaches**(ProfessorA, CourseA)
17. $Professor(ProfessorB)$
18. $Student(ProfessorB)$
19. {ProfessorA} $\not\equiv$ {ProfessorB}
20. {ProfessorB} $\not\equiv$ {StudentA}
21. **takes**(ProfessorB, CourseA)
22. $Student(StudentA)$
23. {ProfessorB} $\not\equiv$ {StudentA}
24. **takes**(StudentA, CourseA)

- C is a consequence map such that $\{\emptyset, \Gamma_o\} \in \mathsf{C}$.

Note that E_{ex} is naïve in the sense that, while it does represent the ontological aspect of O_{ex}, it does not permit any reasoning over the ontological axioms and entities of O_{ex}. In other words, its consequence map cannot be used to derive any theorems aside from the axioms in Γ_o.

For this reason, to accurately represent O_{ex} with its intended reasoning capabilities, it is necessary to re-define it as another extended consequence system with additional, logical tooling. Example 7.4.3 re-defines O_{ex} as an extended consequence system with the reasoning capabilities of $\mathcal{ALC}$. Note that extended consequence systems do not have inherent support for representing role or concept sentences. For this reason, it is necessary to introduce additional symbols to guarantee the syntax for roles and sentences is kept separate. Furthermore, note that $\mathcal{ALC}$ does not have asymmetric, reflexive or inverse roles, but it will serve as a middle step towards fully representing O_{ex} with its intended reasoning capabilities. The formalization of $\mathcal{ALC}$ represented herein is adapted from that presented by Schild [337].

Example 7.4.3 (O_{ex} represented as an extended consequence system.)**.** Consider $E_{ex} = \langle C, \mathsf{C}, C_o, \Gamma_o \rangle$ as defined in example 7.4.2. O_{ex} may be re-defined as an extended consequence system $E_{\mathcal{ALC}} = \langle C', \mathsf{C}', C'_o, \Gamma'_o \rangle$ such that:

- C'_o is a signature where:
  - $C'_{o_1} = C_{o_1} \cup \{\mathsf{concept}, \mathsf{role}\}$
  - $C'_{o_k} = C_{o_k}$, in all other cases.
- C' is a signature such that:
  - $C'_0 = C'_{o_0}$
  - $C'_1 = C'_{o_1} \cup \{\neg\}$
  - $C'_2 = C'_{o_2} \cup \{\forall, \exists, \sqcap, \sqcup\}$
  - $C'_k = \emptyset$ for $k > 2$.
- $\Gamma'_o = \Gamma_o \cup \Gamma_t$, where Γ_t is a set containing the following axioms:

1. $\mathsf{concept}(Person)$
2. $\mathsf{concept}(Professor)$
3. $\mathsf{concept}(Student)$
4. $\mathsf{concept}(\top)$
5. $\mathsf{concept}(\bot)$
6. role(**topObjectProperty**)
7. role(**takes**)
8. role(**teaches**)
9. role(**taughtBy**)

- C' is a consequence map such that $\{\emptyset, \Gamma'_o\} \in \mathsf{C}'$, and such that the following sets are in C':

1. $\{\{\mathsf{concept}(\xi_1), \mathsf{concept}(\xi_2)\}, \{\mathsf{concept}(\xi_1 \sqcap \xi_2), \mathsf{concept}(\xi_1 \sqcup \xi_2)\}\}$
2. $\{\{\mathsf{concept}(\xi_1)\}, \{\mathsf{concept}(\neg\xi_1)\}\}$
3. $\{\{\mathsf{concept}(\xi_1), \mathsf{role}(\xi_2)\}, \{\mathsf{concept}(\forall(\xi_1, \xi_2))\}\}$
4. $\{\{\top \odot \xi_1\}, \{\xi_1\}\}$ and $\{\{\xi_1\}, \{\top \odot \xi_1\}\}$, for $\odot \in \{\sqcap, \sqcup\}$
5. $\{\{\bot \sqcap \xi_1\}, \{\bot\}\}$ and $\{\{\bot\}, \{\bot \sqcap \xi_1\}\}$
6. $\{\{\top \sqcup \xi_1\}, \{\top\}\}$ and $\{\{\top\}, \{\top \sqcup \xi_1\}\}$
7. $\{\{\neg\bot\}, \{\top\}\}$ and $\{\{\top\}, \{\neg\bot\}\}$
8. $\{\{\neg\top\}, \{\bot\}\}$ and $\{\{\bot\}, \{\neg\top\}\}$
9. $\{\{\xi_1, \xi_2\}, \{\xi_1 \sqcap \xi_2\}\}$
   and $\{\{\xi_1 \sqcap \xi_2\}, \{\xi_1, \xi_2\}\}$
10. $\{\{\mathsf{role}(\xi_1), \forall(\xi_1, \top)\}, \{\top\}\}$ and $\{\{\top\}, \{\mathsf{role}(\xi_1), \forall(\xi_1, \top)\}\}$
11. $\{\{\mathsf{concept}(\xi_1), \mathsf{concept}(\xi_2), \mathsf{role}(\xi_3), \forall(\xi_3, \xi_1 \sqcap \xi_2)\}, \{\forall(\xi_3, \xi_1) \sqcap \forall(\xi_3, \xi_2)\}\}$
    and $\{\{\forall(\xi_3, \xi_1) \sqcap \forall(\xi_3, \xi_2)\}, \{\mathsf{concept}(\xi_1), \mathsf{concept}(\xi_2), \mathsf{role}(\xi_3), \forall(\xi_3, \xi_1 \sqcap \xi_2)\}\}$
12. $\{\{\mathsf{role}(\xi_1), \mathsf{concept}(\xi_2), \forall(\xi_1, \xi_2)\}, \{\neg\exists(\xi_1, \neg\xi_2)\}\}$
    and $\{\{\mathsf{role}(\xi_1), \mathsf{concept}(\xi_2), \neg\exists(\xi_1, \neg\xi_2)\}, \{\forall(\xi_1, \xi_2)\}\}$
13. $\{\{\xi_1, \xi_1 \sqsubseteq \xi_2\}, \{\xi_2\}\}$
14. $\{\{\xi_1 \sqsubseteq \xi_2, \xi_2 \sqsubseteq \xi_1\}, \{\xi_1 \equiv \xi_2\}\}$ and $\{\{\xi_1 \equiv \xi_2\}, \{\xi_1 \sqsubseteq \xi_2, \xi_2 \sqsubseteq \xi_1\}\}$
15. $\{\{\neg(\xi_1 \equiv \xi_2)\}, \{\xi_1 \not\equiv \xi_2\}\}$ and $\{\{\xi_1 \not\equiv \xi_2\}, \{\neg(\xi_1 \equiv \xi_2)\}\}$

Intuitively, $E_{\mathcal{ALC}}$ reasoning capabilities come exclusively from its consequence map C', which encodes the axioms of a boolean algebra over $\{\top, \bot, \sqcup, \sqcap, \neg\}$, axioms for reasoning with quantifiers, the modus ponens rule for $\sqsubseteq$, axioms for generating concepts, and additional axioms acting as abbreviations[9]. Notice that the symbols $\forall$ and $\exists$ do not correspond to quantifiers from classical first-order logic. Rather, they are binary symbols with specific axioms dealing with concepts and roles.

Under $E_{\mathcal{ALC}}$, one may state, for instance, that $\vdash_{\mathsf{C}'} Person(ProfessorA)$, since $Professor(ProfessorA)$ and $Professor \sqsubseteq Person$. However, one may note that $\not\vdash_{\mathsf{C}'}$ **taughtBy**$(CourseA, ProfessorA)$, even though $\vdash_{\mathsf{C}'}$ **teaches**$(ProfessorA, CourseA)$.

In order to increase the expressiveness of the extended consequence system representing O_{ex}, it is necessary to add additional axioms pertaining to the asymmetry, irreflexivity and inverse roles. Example 7.4.4 below further extends $E_{\mathcal{ALC}}$ to include such axioms.

Example 7.4.4 (O_{ex} represented as an extended consequence system.). Let $E_{\mathcal{ALC}} = \langle C', \mathsf{C}', C'_o, \Gamma'_o \rangle$ as defined in example 7.4.3. O_{ex} may, once again, be re-defined as an extended consequence system $E_{full} = \langle C^f, \mathsf{C}^f, C^f_o, \Gamma^f_o \rangle$ such that:

- $C^f_o = C'_o$ is a signature

[9] It may be noted that the axiomatization of $\mathcal{ALC}$ within $E_{\mathcal{ALC}}$ is not minimal, in that certain axioms could be removed. Nevertheless, extended consequence systems need not contain the canonical or minimal axiomatization of a logic.

- $C^f = C'$ is a signature
- $\Gamma_o^f = \Gamma_o'$ is a set of axioms
- C^f is a consequence map such that $\mathsf{C}' \subset \mathsf{C}^f$, and such that the following sets are in C^f:

  1. $\{\{\mathsf{role}(\xi_1), \mathsf{concept}(\xi_2), \mathsf{concept}(\xi_3), \text{AsymmetricProperty}(\xi_1), \xi_1(\xi_2, \xi_3)\}, \{\neg\xi_1(\xi_3, \xi_2)\}\}$
  2. $\{\{\mathsf{role}(\xi_1), \mathsf{concept}(\xi_2), \text{IrreflexiveProperty}(\xi_1)\}, \{\neg\xi_1(\xi_2, \xi_2)\}\}$
  3. $\{\{\mathsf{role}(\xi_1), \mathsf{role}(\xi_2), \mathsf{concept}(\xi_3), \mathsf{concept}(\xi_4), \xi_1 \equiv \xi_2^-, \xi_1(\xi_3, \xi_4)\}, \{\xi_2(\xi_4, \xi_3)\}\}$

The extended consequence system E_f is sufficiently expressive to allow reasoning with O_{ex}'s asymmetric, irreflexive and inverse roles. Thus, while in $E_{\mathcal{ALC}}$, one may infer that $\nvdash_{\mathsf{C}'}$ **taughtBy**$(CourseA, ProfessorA)$, it is the case that $\vdash_{\mathsf{C}^f}$ **taughtBy**$(CourseA, ProfessorA)$. Furthermore, it is also the case that $\vdash_{\mathsf{C}^f} \neg$**teaches**$(CourseA, S$

Notice that E_f is substantially more detailed and verbose than O_{ex} as represented in example 7.4.1, including over 50 rules in its consequence map, compared to . However, representing O_{ex} as an extended consequence system allows one to use the da Costian-Tarskianist tools to relate it to over **o**ntologies represented in the same manner. Furthermore, one may use da Costian-Tarskianist decomposition to split O_{ex} into two or more separate **o**ntologies, such as an **o**ntology pertaining to students and courses, and another pertaining to professors and courses.

Due to the verbosity and mechanical nature of the task, this section will not present a fully detailed extended consequence system representing a top-level **o**ntology. Appendix B presents the "intermediate" representation (as done in example 7.4.1) of BFO, which may be used to generate an extended consequence system.

The step-wise process to generate O_{ex}'s extended consequence system representation with fully expressive capabilities may be generalized to generate extended consequence systems for any **o**ntology. The steps below informally describe how this procedure may be generalized.

Step 1: Given an **o**ntology O represented in a language L, identify its symbols, axioms, individuals (if any), and intended logic **L** of representation;

Step 2: Start with an extended consequence system $E = \langle C, \mathsf{C}, C_o, \Gamma_o \rangle$ such that its signatures, consequence map and ontological axioms are empty;

Step 3: For each symbol s in O:

Substep 1: Add s to C and C_o;

Substep 2: If the intended logic **L** is a description logic, identify if s is a concept or role: if so, add an axiom to Γ_o which states the respective case;

Step 4: For each individual i in O:

Substep 1: Add i to C_{o_0} of C_o;

Step 5: For each axiom a in O:

Substep 1: Add a to Γ_o;

Step 6: Add $\{\emptyset, \Gamma_o\}$ to C;

Step 7: If the intended logic **L** is a description logic:

Substep 1: Add rules for the syntax of concepts and roles to C;

Substep 2: Add rules for role and concept constructors to C;

Step 8: Add the axiomatization of **L** to C.

Step 9: The resulting extended consequence system E should represent O with the intended expressiveness of **L**.

It is not in scope for this work to present the algorithmic details of the procedure, a task that may be deferred to future research. However, the informal procedure may serve as the basis of formalized, robust and efficient algorithms, with better understood computational complexity. Additionally, the procedure is "one-way" in the sense that it does not provide means to represent an extended consequence system E in representation language L. While this task is also deferred to future works on the matter, developing a mapping between representation languages and extended consequence systems is one of the first steps towards a concrete implementation of da Costian-Tarskianism and may thus lead to fruitful results.

Chapter 8

Conclusion

At the heart of this essay lies a fundamental problem, which can be broadly conceptualized as the question of how to reason about what there exists. Traditionally, Logic is taken to be the field of Philosophy concerned with "how to reason", whereas Ontology attempts to answer "what there exists". Thus, it is by no coincidence that this essay sits at the intersection of these two fields, and draws from many other fields such as Computer Science and Artificial Intelligence. This essay posits that there is no unique answer to the fundamental problem and that one needs to manage the diversity of possible answers. This diversity of possible answers is what leads to what is called "ontological heterogeneity", hence the essay's title. Furthermore, the da Costian-Tarskianist framework of chapter 6 serves as a novel way to manage ontological heterogeneity and implement ontological interoperability — the ability to effectively operate on different ontologies.

However, an initial question one may inquire is how exactly Ontology and Logic came to be intersected. This is the goal of chapter 2, which summarizes the history of Ontology up until the birth of Applied Ontology. Despite its name, Applied Ontology does not refer to applying philosophy ontological inquiry to daily situations. It refers to a sub-area of Ontology where ontologies are seen as objects, much like logics are seen as objects in the broader area of Logic. By presenting Guarino's definition of ontology, chapter 2 delimits the scope of this essay to relationship between Logic and the relatively new Applied Ontology. **O**ntologies, with bold-face **O**, are then understood as possible ways of representing what there is, given certain ontological commitments formalized as axioms of a logic.

Chapter 3 examines what structure such **o**ntologies may have. Historically, Ontology as an area is concerned with the structure of what there is as a whole. However, this is not necessarily the case for **o**ntologies. Chapter 3 makes it clear by presenting an extended version of Guarino's taxonomy of **o**ntologies and discussing at length about instances of **o**ntologies whose objective is to represent very fundamental concepts, top-level **o**ntologies such as BFO, UFO and DOLCE, and **o**ntologies attempting to represent very specific domains, such as oil industry and biology **o**ntologies.

The diversity of existing **o**ntologies, even at the top-level, is one of the empirical arguments employed by this essay to argue that the fundamental problem has no unique answer.Chapter 4 expands on the problem by briefly discussing its history and formalizing it as the MOP. The chapter espouses a reduction thesis by arguing that Guarino's definition and all **o**ntologies presented in chapter 3 are represented, either directly or indirectly, through logics. This reduction thesis, thus, leads to a specialization of the problem called AMOP, the Applied Meta-Ontological Problem.

Chapter 4 argues that in order to address the AMOP, one should accept logical pluralism. The arguments are not only empirical but also pragmatic — there exists ontological diversity in the literature, and in real-world applications it is necessary to accept different **o**ntologies. However, as it is expected from philosophical inquiry, this leads to further questions: what type pluralism should one accept and how this choice may lead to actual ontological heterogeneity or interoperability. The purpose of 5 is to expand on these questions.

By briefly presenting different kinds of logical pluralism, chapter 5 attempts to clarify how one specific kind of pluralism may effectively lead to an heterogeneous framework. One of the existing frameworks in the literature is Lücke's Carnapian-Goguenism [235], drawing from Carnap's Principle of Tolerance and Goguen and Burstall's institution theory. Chapter 5 presents the theory of Carnapian-Goguenism, which again re-defines what is an **o**ntology. The chapter discusses some of its limitations, both technical and fundamental in nature. Regarding the former, Carnapian-Goguenism relies on flattening **o**ntologies as a single **o**ntology to allow for interoperability, a computationally expensive operation. Regarding the latter, Carnapian-Goguenism is fundamentally rooted on the fact that **o**ntologies should be represented by their semantics — specifically, by set-theoretical semantics.

Heavily drawing from Carnapian-Goguenism and in light of its limitations, chapter 6 presents a novel framework for ontological heterogeneity based on one of the pluralist views discussed in chapter 5. Named da Costian-Tarskianism, it is based on da Costa's Principle of Non-Triviality and Tarski's consequence operators. By representing **o**ntologies as an extension of consequence systems, unsurprisingly named extended consequence systems, da Costian-Tarskianism is not limited to set-theoretical semantics. Additionally, while Carnapian-Goguenism supports three different ontological operations, refinement, integration, and connection, da Costian-Tarskianism supports an additional ontological operation ontological decomposition. Nevertheless, as chapter 6 notes, da Costian-Tarskianism does not yet enjoy an implementation such as the Carnapian-Goguenist HETS.

Carnapian-Goguenism and da Costian-Tarskianism are able to operate over **o**ntologies based on logics. However, they are not able to determine what logic should be used to represent an **o**ntology. Chapter 7 discusses the question of what exactly makes a logic a good candidate for ontological representation by referring back to the representation languages of chapter 5. In general, **o**ntologies are represented by languages derived from description logics due to their computational characteristics, such as decidability and complexity of reasoning. Nonetheless, as it is noted in chapter 7, some of the real-world challenges of building **o**ntologies cannot be addressed purely by traditional description logics and often require additional tooling.

Chapter 7 presents a possible paraconsistent extension of description logics, originally devised by W. Carnielli, M. E. Coniglio, and Bueno-Soler [73], to potentially address the challenges of inconsistency-tolerant reasoning and open-world *versus* closed-world assumptions. The logic $\mathcal{ALC}_{\mathcal{LR}}$ extends the logic $\mathcal{ALC}$ with a consistency operator, which may be useful in the context of ontological integration where information may lead to inconsistencies. Chapter 7 also sketches how one may effectively transform an **o**ntology represented by a representation language, such as RDF, into an extended consequence system. This transformation represents the first step towards building an automated system implementing da Costian-Tarskianism.

As might have been anticipated, attempting to provide an answer to a number of questions has led to the emergence of a number of questions, which may be addressed as future work. What follows is a brief discussion of a few potential venues for further research.

Strengthening da Costian-Tarskianism

Currently, da Costian-Tarskianism is lacking in several aspects. Firstly, it lacks a robust theory of compositionality. It is not clear how the ontological operations may compose and whether applying any number of ontological operations (e.g. integrating, then refining and connecting) induces a well-defined structure, such as an induced algebra. Therefore, it may be fruitful to further develop da Costian-Tarskianism in this sense.

Additionally, it is conjectured that extended consequence systems are isomorphic to consequence systems. Further developing the relationship between the categories of these two objects may allow for easier development of da Costian-Tarskianism, as the theory of consequence systems is well-developed and contains a plethora of results. As discussed in chapter 6, algebraic fibring is also not the only operation for combining logics and expanding da Costian-Tarskianism to include other mechanisms of combination is a clear path for further research.

It may be noted that (extended) consequence systems are not capable of representing first-order logic or any quantifiers for the matter. W. Carnielli et al. [74] present first-order and higher-order logic systems to address this limitation. While chapter 6 did not discuss such concepts, they may be incorporated into da Costian-Tarskianism as a way to overcome this limitation.

Lastly, da Costian-Tarskianism does not enjoy an implementation akin to the Carnapian-Goguenist Hets. Therefore, building an implementation is clearly a potential direction of research. As proposed in chapter 6, the language Julia may be used to implement the category-theoretical aspect of extended consequence system. However, it may be beneficial to integrate such implementation into an already existing and larger project, such as Apache Jena[1], or to directly ensure the implementation is compatible with the OWL API [182].

Neurosymbolic Artificial Intelligence

Very recently, Neurosymbolic Artificial Intelligence has become one of the main research trends at the intersection of Applied Ontology and Artificial Intelligence. Neurosymbolic Artifical Intelligence combines **o**ntologies with machine learning models, which leverage probability theory to learn the patterns and connections between quite large data sets of words or images. For instance, widely popular Large Language Models, or LLMs, are a specific type of machine learning models derived from the transformer deep learning ar-

[1]Available at https://jena.apache.org/.

chitecture introduced by Vaswani et al. [390]. LLMs can be queried to generate a response based on inputs or "prompts", a process that is exemplified by OpenAI's ChatGPT, thus being classified within the domain of Generative Artificial Intelligence.

Although this essay did not aim to address Neurosymbolic Artificial Intelligence directly, it stands as a very promising research direction, being the main topic of 2024's edition of the Ontology Summit[2] — an annual series of events co-organized by Ontolog and several national US-based research centers, such as US National Institute of Standards and Technology (NIST).

As argued by Neuhaus [274], LLMs or any machine learning model do not replace **o**ntologies or fully automate **o**ntology development. Neuhaus [274] argues that machine learning models and **o**ntologies have differing use cases: in particular, LLMs are very useful at navigating ambiguities or different perspectives in data, but they do not persist any ontological commitments, nor are trained to resolve ambiguities or provide logically consistent responses. Furthermore, there is no underlying **o**ntology that can be extracted from an LLM by prompting it. Additionally, Neuhaus [274] argues that **o**ntology development cannot be *fully* automated by machine learning models because **o**ntologies require consensus, as discussed in chapter 2, and because, as argued in chapter 4, there is no *one* **o**ntology that can represent a domain.

Nevertheless, the research directions in Neurosymbolic Artificial Intelligence that are related to **o**ntologies can be broadly classified into two categories: firstly, the use of machine learning models to facilitate **o**ntology development; and secondly, the incorporation of ontological information into machine learning based systems. Although **o**ntology development cannot be fully automated, some researchers such as Babaei Giglou, D'Souza, and Auer [24] and Lopes et al. [232] have been developing tools based on LLMs to aid the **o**ntology development process. On the matter of incorporating ontological reasoning in machine learning based systems, Gaur and Sheth [131], Jaimini, Henson, and Sheth [197], Marcus [246], Mossakowski [262], Sheth [352], Sheth, Roy, and Gaur [353], and J. Sowa [365] have been conducting both theoretical and applied research to both integrate machine learning inference with ontological information, and to use ontological reasoning to explain machine learning inference. One key difference between **o**ntologies and machine learning models is that, in general, the latter works as a "black-box" — that is, there is no possibility of explaining why a model inferred a particular result. The concept of explanability has garnered much attention in the literature recently and exploring it is a potential venue for further philosophical research.

Relating back to the scope of this essay, integrating da Costian-Tarskianism or Carnapian-Goguenism into a larger framework allowing for neurosymbolic reasoning may be a fruitful research direction. Additionally, developing an abstract theory of neurosymbolic reasoning using category-theoretical descriptions might prove to be useful to better characterize how **o**ntologies and machine learning models may be combined abstractly. For instance, one may define a category **LLM** representing a particular LLM, such that its objects are strings of text and its morphisms correpond to "prompts". By leveraging the existing literature on institution theory and extended consequence systems, one may category-theoretically relate LLMs and logics. In any case, there is much work to be done in any effort one chooses to undertake to advance Neurosymbolic Artificial Intelligence.

[2]Details available at https://ontologforum.com/index.php/OntologySummit2024.

Closing Thoughts

Ontology and Logic are arguably two of the oldest disciplines of human knowledge and, despite their age, there is no foreseeable shortage of matters to discuss. This essay attempts to contribute to the discussion by recasting one of the oldest problems of reasoning under a more recent and pragmatic light, and by describing one possible answer to the problem. In doing so, the author hopes that the findings herewithin may be used by future philosophers and researchers to keep the discussion alive.

Bibliography

[1] M. Adach, K. Hänninen, and K. Lundqvist. "A Combined Security Ontology based on the Unified Foundational Ontology." In: *2022 IEEE 16th International Conference on Semantic Computing (ICSC)*. 2022, pp. 187–194. DOI: 10.1109/ICSC52841.2022.00039.

[2] N. Adamovic et al. *OntoCommons Roadmap V1*. Jan. 2023. DOI: 10.5281/zenodo.7544509. URL: https://doi.org/10.5281/zenodo.7544509.

[3] K. Ajdukiewicz. "Die syntaktische konnexitat." In: *Studia philosophica* (1935), pp. 1–27.

[4] L. Albertazzi. "Formal and Material Ontology." In: *Formal Ontology*. Ed. by R. Poli and P. Simons. Dordrecht: Springer Netherlands, 1996, pp. 199–232. ISBN: 978-94-015-8733-4. DOI: 10.1007/978-94-015-8733-4_8. URL: https://doi.org/10.1007/978-94-015-8733-4_8.

[5] L. Albertazzi. *Shapes of Forms: From Gestalt psychology and phenomenology to ontology and mathematics*. Vol. 275. Springer Science & Business Media, 1999.

[6] J. F. Allen. "Maintaining Knowledge about Temporal Intervals." In: *Commun. ACM* 26.11 (Nov. 1983), pp. 832–843. ISSN: 0001-0782. DOI: 10.1145/182.358434. URL: https://doi.org/10.1145/182.358434.

[7] C. Alrabbaa et al. "On the Eve of True Explainability for OWL Ontologies: Description Logic Proofs with Evee and Evonne." English. In: *DL 2022 - Proceedings of the 35th International Workshop on Description Logics, co-located with Federated Logic Conference, FLoC 2022*. Ed. by O. Arieli et al. Vol. 3263. CEUR Workshop Proceedings. 35th International Workshop on Description Logics, DL 2022 ; Conference date: 07-08-2022 Through 10-08-2022. CEUR-WS, 2022.

[8] H. Andréka, I. Németi, and J. van Benthem. "Modal Languages and Bounded Fragments of Predicate Logic." In: *Journal of Philosophical Logic* 27.3 (1998), pp. 217–274. ISSN: 00223611, 15730433. URL: http://www.jstor.org/stable/30226645 (visited on 03/13/2024).

[9] P. B. Andrews. *An introduction to mathematical logic and type theory: to truth through proof*. Vol. 27. Springer Science & Business Media, 2013.

[10] T. Aquinas. "Thomas Aquinas's Commentary on Aristotle's "metaphysics"." In: ed. by J. P. Rowan. Aristotelian Commentary S. Indiana, PA: Dumb Ox Books, Nov. 1995.

[11] R. W. Arroyo and G. O. da Silva. "Taking Models Seriously and Being a Linguistic Realist." In: *Principia: An International Journal of Epistemology* 26.1 (2022), pp. 73–94. DOI: 10.5007/1808-1711.2022.e84309.

[12] A. Artale, E. Franconi, et al. "Temporal Description Logics." In: *Handbook of Temporal Reasoning in Artificial Intelligence* 1 (2005), pp. 375–388.

[13] A. Artale et al. "Living without Beth and Craig: Definitions and Interpolants in Description and Modal Logics with Nominals and Role Inclusions." In: *ACM Trans. Comput. Logic* 24.4 (Oct. 2023). ISSN: 1529-3785. DOI: 10.1145/3597301. URL: https://doi.org/10.1145/3597301.

[14] M. Ashburner et al. "Gene ontology: tool for the unification of biology." In: *Nature genetics* 25.1 (2000), pp. 25–29.

[15] M. N. Asim et al. "A survey of ontology learning techniques and applications." In: *Database* 2018 (Oct. 2018), bay101. ISSN: 1758-0463. DOI: 10.1093/database/bay101. eprint: https://academic.oup.com/database/article-pdf/doi/10.1093/database/bay101/27329264/bay101.pdf. URL: https://doi.org/10.1093/database/bay101.

[16] E. Astesiano, H. Kreowski, and B. Krieg-Brückner. *Algebraic Foundations of Systems Specification*. IFIP State-of-the-Art Reports. Springer Berlin Heidelberg, 2012. ISBN: 9783642598517. URL: https://books.google.com.br/books?id=MtmpCAAAQBAJ.

[17] F. Baader, ed. *Description Logic Handbook*. Cambridge University Press, 2003.

[18] F. Baader and O. F. Gil. "Extending the description logic EL with threshold concepts induced by concept measures." In: *Artificial Intelligence* 326 (2024), p. 104034.

[19] F. Baader and B. Hollunder. "A terminological knowledge representation system with complete inference algorithms." In: *Processing Declarative Knowledge*. Ed. by H. Boley and M. M. Richter. Berlin, Heidelberg: Springer Berlin Heidelberg, 1991, pp. 67–86. ISBN: 978-3-540-46667-3.

[20] F. Baader and C. Lutz. "13 Description logic." In: *Studies in Logic and Practical Reasoning*. Vol. 3. Elsevier, 2007, pp. 757–819.

[21] F. Baader, C. Lutz, and B. Suntisrivaraporn. "CEL — A Polynomial-Time Reasoner for Life Science Ontologies." In: *Automated Reasoning*. Ed. by U. Furbach and N. Shankar. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 287–291. ISBN: 978-3-540-37188-5.

[22] F. Baader and U. Sattler. "An Overview of Tableau Algorithms for Description Logics." In: *Studia Logica* 69.1 (Oct. 2001), pp. 5–40. ISSN: 1572-8730. DOI: 10.1023/A:1013882326814. URL: https://doi.org/10.1023/A:1013882326814.

[23] F. Baader et al. *The Description Logic Handbook: Theory, Implementation and Applications*. 2nd ed. Cambridge University Press, 2007. DOI: 10.1017/CBO9780511711787.

[24] H. Babaei Giglou, J. D'Souza, and S. Auer. "LLMs4OL: Large Language Models for Ontology Learning." In: *The Semantic Web – ISWC 2023*. Ed. by T. R. Payne et al. Cham: Springer Nature Switzerland, 2023, pp. 408–427. ISBN: 978-3-031-47240-4.

[25] K. Baclawski et al. "Ontology summit 2019 communiqué: Explanations." In: *Applied Ontology* 15.1 (2020), pp. 91–107.

[26] J. A. Bateman et al. "GUM: The Generalized Upper Model." In: *Appl. Ontol.* 17.1 (Jan. 2022), pp. 107–141. ISSN: 1570-5838. DOI: 10.3233/AO-210258. URL: https://doi.org/10.3233/AO-210258.

[27] V. Bazhanov. "The imaginary geometry of N. I. Lobachevsky and the imaginary logic of N. A. Vasiliev." In: *The Review of Modern Logic* 4 (Apr. 1994).

[28] V. A. Bazhanov. "The dawn of paraconsistency: Russia's logical thought in the turn of XX century." In: *Manuscrito* 34 (2011), pp. 89–98.

[29] J. C. Beall and G. Restall. "Defending Logical Pluralism." In: *Logical Consequence: Rival Approaches*. Hermes, 2001, pp. 1–22.

[30] J. Beall and G. Restall. "Logical Pluralism." In: *Australasian Journal of Philosophy* 78.4 (2000), pp. 475–493. DOI: 10.1080/00048400012349751.

[31] J. Beall and G. Restall. *Logical Pluralism*. Ed. by G. Restall. Oxford, England: Oxford University Press, 2005.

[32] C. I. Beckwith. *Greek Buddha*. Princeton, NJ: Princeton University Press, June 2015.

[33] N. Belnap. "How a Computer Should Think." In: *Contemporary aspects of philosophy*. Ed. by G. Ryle. Oriel Press, 1977.

[34] N. D. Belnap. "A Useful Four-Valued Logic." In: *Modern Uses of Multiple-Valued Logic*. Ed. by J. M. Dunn and G. Epstein. Dordrecht: Springer Netherlands, 1977, pp. 5–37. ISBN: 978-94-010-1161-7. DOI: 10.1007/978-94-010-1161-7_2. URL: https://doi.org/10.1007/978-94-010-1161-7_2.

[35] J. Berg. *Ontology Without Ultrafilters and Possible Worlds: An Examination of Bolzano's Ontology*. Academia, 1992.

[36] E. W. Beth. *Semantic Entailment and Formal Derivability*. Noord-Hollandsche, 1955.

[37] J. Béziau, ed. *Logica Universalis: Towards a General Theory of Logic*. Boston: Birkhäuser Basel, 2007.

[38] M. Bidoit and P. D. Mosses. "1 Introduction." In: *CASL User Manual: Introduction to Using the Common Algebraic Specification Language*. Ed. by M. Bidoit and P. D. Mosses. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 3–9. ISBN: 978-3-540-44397-1. DOI: 10.1007/978-3-540-44397-1_1. URL: https://doi.org/10.1007/978-3-540-44397-1_1.

[39] M. Bienvenu. "A short survey on inconsistency handling in ontology-mediated query answering." In: *KI-Künstliche Intelligenz* 34.4 (2020), pp. 443–451.

[40] B. L. Bihan and A. Barton. "Analytic Metaphysics Versus Naturalized Metaphysics: The Relevance of Applied Ontology." In: *Erkenntnis* 86.1 (2021), pp. 21–37. DOI: 10.1007/s10670-018-0091-8.

[41] F. Bobillo and U. Straccia. "fuzzyDL: An expressive fuzzy description logic reasoner." In: *2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence)*. IEEE. 2008, pp. 923–930.

[42] A. Borgida and P. F. Patel-Schneider. "A semantics and complete algorithm for subsumption in the classic description logic." In: *J. Artif. Int. Res.* 1.1 (June 1994), pp. 277–308. ISSN: 1076-9757.

[43] A. Borgida and L. Serafini. "Distributed description logics: Assimilating information from peer sources." In: *Journal on data semantics I* (2003), pp. 153–184.

[44] S. Borgo et al. "DOLCE: A descriptive ontology for linguistic and cognitive engineering." In: *Applied ontology* 17.1 (2022), pp. 45–69.

[45] S. Borgo et al. "Foundational Ontologies in Action." In: *Appl. Ontol.* 17.1 (Jan. 2022), pp. 1–16. ISSN: 1570-5838. DOI: 10.3233/AO-220265. URL: https://doi.org/10.3233/AO-220265.

[46] W. N. Borst. "Construction of Engineering Ontologies for Knowledge Sharing and Reuse." PhD thesis. Enschede, Sept. 1997. URL: http://doc.utwente.nl/17864/.

[47] L. BOTHA, T. MEYER, and R. PEÑALOZA. "The Probabilistic Description Logic." In: *Theory and Practice of Logic Programming* 21.4 (2021), pp. 404–427. DOI: 10.1017/S1471068420000460.

[48] A. Bouras, P. Gouvas, and G. Mentzas. "ENIO: An Enterprise Application Integration Ontology." In: *18th International Workshop on Database and Expert Systems Applications (DEXA 2007)*. 2007, pp. 419–423. DOI: 10.1109/DEXA.2007.25.

[49] G. Bourguin et al. "Towards Ontologically Explainable Classifiers." In: *Artificial Neural Networks and Machine Learning – ICANN 2021*. Ed. by I. Farkaš et al. Cham: Springer International Publishing, 2021, pp. 472–484. ISBN: 978-3-030-86340-1.

[50] G. E. P. Box. "Science and Statistics." In: *Journal of the American Statistical Association* 71.356 (1976), pp. 791–799. DOI: 10.1080/01621459.1976.10480949. eprint: https://www.tandfonline.com/doi/pdf/10.1080/01621459.1976.10480949. URL: https://www.tandfonline.com/doi/abs/10.1080/01621459.1976.10480949.

[51] R. J. Brachman. *A Structural Paradigm for Representing Knowledge.* Tech. rep. BOLT BERANEK and NEWMAN INC CAMBRIDGE MASS, 1978.

[52] R. J. Brachman et al. ""Reducing" classic to practice: Knowledge representation theory meets reality." In: *Artificial Intelligence* 114.1 (1999), pp. 203–237. ISSN: 0004-3702. DOI: https://doi.org/10.1016/S0004-3702(99)00078-8. URL: https://www.sciencedirect.com/science/article/pii/S0004370299000788.

[53] R. J. Brachman and H. J. Levesque. "The tractability of subsumption in frame-based description languages." In: *Proceedings of the Fourth AAAI Conference on Artificial Intelligence*. AAAI'84. Austin, Texas: AAAI Press, 1984, pp. 34–37.

[54] R. J. Brachman and J. G. Schmolze. "An Overview of the KL-ONE Knowledge Representation System." In: *Cognitive Science* 9.2 (1985), pp. 171–216. DOI: 10.1207/s15516709cog0902_1.

[55] G. Brewka, I. Niemelä, and M. Truszczyński. "Nonmonotonic reasoning." In: *Foundations of Artificial Intelligence* 3 (2008), pp. 239–284.

[56] M. Buchheit, F. M. Donini, and A. Schaerf. "Decidable reasoning in terminological knowledge representation systems." In: *Journal of artificial intelligence research* 1 (1993), pp. 109–138.

[57] P. Buitelaar, D. Olejnik, and M. Sintek. "A protégé plug-in for ontology extraction from text based on linguistic analysis." In: *European Semantic Web Symposium*. Springer. 2004, pp. 31–44.

[58] T. Bylander and B. Chandrasekaran. "Generic tasks for knowledge-based reasoning: the "right" level of abstraction for knowledge acquisition." In: *International Journal of Man-Machine Studies* 26.2 (1987), pp. 231–243. ISSN: 0020-7373. DOI: https://doi.org/10.1016/S0020-7373(87)80093-7. URL: https://www.sciencedirect.com/science/article/pii/S0020737387800937.

[59] E. G. Caldarola and A. M. Rinaldi. "An approach to ontology integration for ontology reuse." In: *2016 IEEE 17th international conference on information reuse and integration (IRI)*. IEEE. 2016, pp. 384–393.

[60] C. Caleiro and S. Marcelino. "Decidability Preservation and Complexity Bounds for Combined Logics." In: *Mathematics* 10.19 (2022). ISSN: 2227-7390. DOI: `10.3390/math10193481`. URL: `https://www.mdpi.com/2227-7390/10/19/3481`.

[61] C. Caleiro and J. Ramos. "From Fibring to Cryptofibring. A Solution to the Collapsing Problem." In: *Logica Universalis* 1 (Jan. 2007), pp. 71–92. DOI: `10.1007/s11787-006-0004-5`.

[62] D. Calvanese. "Finite Model Reasoning in Description Logics." In: *KR* 96 (1996), pp. 292–303.

[63] D. Calvanese, G. De Giacomo, and M. Lenzerini. "A Framework for Ontology Integration." In: *The emerging semantic web*. 2001, pp. 201–214.

[64] D. Calvanese, G. De Giacomo, and M. Lenzerini. "On the decidability of query containment under constraints." In: *Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*. 1998, pp. 149–158.

[65] C. R. Caret. "Why Logical Pluralism?" In: *Synthese* 198.Suppl 20 (2019), pp. 4947–4968. DOI: `10.1007/s11229-019-02132-w`.

[66] C. R. Caret and T. K. Kissel. "Pluralistic Perspectives on Logic: An Introduction." In: *Synthese* 198.Suppl 20 (2020), pp. 4789–4800. DOI: `10.1007/s11229-019-02525-x`.

[67] R. Carnap. "Empiricism, Semantics and Ontology." In: *Revue Internationale de Philosophie* 4.11 (1950), pp. 20–40.

[68] R. Carnap. *Introduction to Semantics*. Cambridge: Harvard University Press, 1942.

[69] R. Carnap. *Meaning and Necessity: A Study in Semantics and Modal Logic*. Chicago, IL, USA: University of Chicago Press, 1947.

[70] R. Carnap. *The Logical Structure of the World*. Berkeley: University of California Press, 1967.

[71] R. Carnap. *The Logical Syntax of Language*. London: K. Paul, Trench, Trubner & Co., 1937.

[72] W. Carnielli, M. E. Coniglio, and D. Fuenmayor. "Logics of Formal Inconsistency enriched with replacement: an algebraic and modal account." In: *The Review of Symbolic Logic* 15.3 (2022), pp. 771–806.

[73] W. Carnielli, M. E. Coniglio, and J. Bueno-Soler. "On $\mathcal{PALC}$: Description logics over Logics of Formal Inconsistency with replacement." In: (forthcoming).

[74] W. Carnielli et al. *Analysis and Synthesis of Logics: how to cut and paste reasoning systems*. Vol. 35. Springer Science & Business Media, 2008.

[75] W. A. Carnielli and J. Marcos. "A taxonomy of C-systems." In: *Paraconsistency*. CRC Press, 2002, pp. 1–94.

[76] W. A. Carnielli. "Systematization of Finite Many-Valued Logics Through the Method of Tableaux." In: *The Journal of Symbolic Logic* 52.2 (1987), pp. 473–493. ISSN: 00224812. URL: http://www.jstor.org/stable/2274395 (visited on 02/24/2024).

[77] W. A. Carnielli, C. Pizzi, and J. Bueno-Soler. *Modalities and Multimodalities.* Dordrecht, Netherland: Springer, 2008.

[78] A. Chakravartty. *Scientific Ontology: Integrating Naturalized Metaphysics and Voluntarist Epistemology.* Oxford University Press, Aug. 2017. ISBN: 9780190651459. DOI: 10.1093/oso/9780190651459.001.0001. URL: https://doi.org/10.1093/oso/9780190651459.001.0001.

[79] D. Chalmers, D. Manley, and R. Wasserman. *Metametaphysics: New essays on the foundations of ontology.* OUP Oxford, 2009.

[80] B. F. Chellas. *Modal Logic: An Introduction.* New York: Cambridge University Press, 1980.

[81] A. Church. "An Unsolvable Problem of Elementary Number Theory." In: *Journal of Symbolic Logic* 1.2 (1936), pp. 73–74. DOI: 10.2307/2268571.

[82] A. Citkin and A. Muravitsky. *Consequence Relations: An Introduction to the Lindenbaum-Tarski Method.* Oxford University Press, 2022.

[83] N. B. Cocchiarella. *Formal Ontology and Conceptual Realism.* Dordrecht, Netherland: Springer, 2007.

[84] N. B. Cocchiarella. "Formal Ontology and the Foundations of Mathematics." In: *Bertrand Russell's Philosophy.* Ed. by G. Nakhnikian. London: Duckworth, 1974.

[85] N. B. Cocchiarella. "Logic and Ontology." In: *Axiomathes* 12.1-2 (2001), pp. 117–150. DOI: 10.1023/a:1012758003706.

[86] R. Cockett and R. Seely. "Polarized category theory, modules, and game semantics." In: *Theory and Applications of Categories* 18.2 (2007), pp. 4–101.

[87] J. A. Coffa. *The Semantic Tradition from Kant to Carnap: To the Vienna Station.* Cambridge University Press, 1991. DOI: 10.1017/CBO9781139172240.

[88] M. E. Coniglio, A. Sernadas, and C. Sernadas. "Preservation by fibring of the finite model property." In: *Journal of Logic and Computation* 21.2 (July 2010), pp. 375–402. ISSN: 0955-792X. DOI: 10.1093/logcom/exq022. eprint: https://academic.oup.com/logcom/article-pdf/21/2/375/2775948/exq022.pdf. URL: https://doi.org/10.1093/logcom/exq022.

[89] M. Coniglio. "Towards a Stronger Notion of Translation Between Logics." In: *Manuscrito* 28.2 (2005), pp. 231–262.

[90] M. E. Coniglio, F. D. C. Luis, and M. P. Newton. "Modal logic with non-deterministic semantics: Part I—Propositional case." In: *Logic Journal of the IGPL* 28.3 (Sept. 2019), pp. 281–315. ISSN: 1367-0751. DOI: 10.1093/jigpal/jzz027. eprint: https://academic.oup.com/jigpal/article-pdf/28/3/281/33330559/jzz027.pdf. URL: https://doi.org/10.1093/jigpal/jzz027.

[91] G. O. Consortium. "The Gene Ontology (GO) database and informatics resource." In: *Nucleic acids research* 32.suppl_1 (2004), pp. D258–D261.

[92] G. O. Consortium. "The gene ontology resource: 20 years and still GOing strong." In: *Nucleic acids research* 47.D1 (2019), pp. D330–D338.

[93] R. T. Cook. "Let a Thousand Flowers Bloom: A Tour of Logical Pluralism." In: *Philosophy Compass* 5.6 (2010), pp. 492–504. DOI: 10.1111/j.1747-9991.2010.00286.x.

[94] O. Corcho, M. Fernández-López, and A. Gómez-Pérez. "Methodologies, tools and languages for building ontologies. Where is their meeting point?" In: *Data & Knowledge Engineering* 46.1 (2003), pp. 41–64. ISSN: 0169-023X. DOI: https://doi.org/10.1016/S0169-023X(02)00195-7. URL: https://www.sciencedirect.com/science/article/pii/S0169023X02001957.

[95] O. Corcho and A. Gómez-Pérez. "A roadmap to ontology specification languages." In: *International Conference on Knowledge Engineering and Knowledge Management*. Springer. 2000, pp. 80–96.

[96] N. Cordero. *By Being, It Is: The Thesis of Parmenides*. Parmenides, 2004.

[97] N. C. A. da Costa. "Logic and Ontology." In: *Principia: An International Journal of Epistemology* 6.2 (2002), pp. 179–298.

[98] N. C. A. da Costa. "On the Theory of Inconsistent Formal Systems." In: *Anuário da Sociedade Brasileira de Matemática* 2.2 (1958), pp. 16–19.

[99] N. C. A. da Costa. "On the Theory of Inconsistent Formal Systems." In: *Notre Dame Journal of Formal Logic* 15.4 (1974), pp. 497–510. DOI: 10.1305/ndjfl/1093891487.

[100] N. C. A. da Costa. "Sistemas formais inconsistentes." PhD thesis. Curitiba, Mar. 1963.

[101] J. Cottingham, R. Stoothoff, and D. Murdoch. *The Philosophical Writings of Descartes: Volume 2*. Cambridge University Press, 1984.

[102] J. Courtine. *Suarez Et le Système de la Métaphysique*. Presses Universitaires de France - PUF, 1990.

[103] I. M. Crombie. *An Examination of Plato's Doctrines Vol 2 : Volume 2 Plato on Knowledge and Reality*. Routledge, 1963.

[104] Cycorp. *Technology Overview*. Tech. rep. Cycorp, 2021.

[105] I. M. L. D'Ottaviano and E. L. Gomes. "Considerações sobre o desenvolvimento da lógica no Brasil." In: *CLE e-Prints (Online)* 11 (2011), pp. 1–50.

[106] I. M. L. D'Ottaviano and C. Hifume. "Peircean Pragmatic Truth and da Costa's Quasi-Truth." In: *Model-Based Reasoning in Science, Technology, and Medicine*. Ed. by L. Magnani and P. Li. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 383–398. ISBN: 978-3-540-71986-1. DOI: 10.1007/978-3-540-71986-1_22. URL: https://doi.org/10.1007/978-3-540-71986-1_22.

[107] T. Dalmonte et al. "Non-Normal Modal Description Logics." In: *Logics in Artificial Intelligence*. Ed. by S. Gaggl, M. V. Martinez, and M. Ortiz. Cham: Springer Nature Switzerland, 2023, pp. 306–321. ISBN: 978-3-031-43619-2.

[108] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order*. 2nd ed. Cambridge University Press, 2002. DOI: 10.1017/CBO9780511809088.

[109] E. Davis. "Naive Physics Perplex." In: *AI Mag.* 19 (1997), pp. 51–79.

[110] A. De Baas et al. "Review and Alignment of Domain-Level Ontologies for Materials Science." In: *IEEE Access* 11 (2023), pp. 120372–120401. DOI: 10.1109/ACCESS.2023.3327725.

[111] J. Dejnožka. "Being Qua Identity in Russell's Ontologies." In: *Theory and History of Ontology, Raul Corazzon's ontology site* (2018).

[112] R. Descartes. "Meditations on first philosophy." In: *Seven Masterpieces of Philosophy*. Routledge, 2016, pp. 63–108.

[113] R. Diaconescu. "Grothendieck Institutions." In: *Applied Categorical Structures* 10 (Aug. 2002), pp. 383–402. DOI: 10.1023/A:1016330812768.

[114] R. Diaconescu. *Institution-Independent Model Theory*. Basel, Switzerland: Birkhäuser, 2008.

[115] J. Duck-Mayr. "Explaining legal inconsistency." In: *Journal of Theoretical Politics* 34.1 (2022), pp. 107–126.

[116] P. Duhem. *The Aim and Structure of Physical Theory*. Vol. 126. Princeton University Press, 1982. ISBN: 9780691025247. URL: http://www.jstor.org/stable/j.ctv1nj34vm (visited on 05/20/2024).

[117] J. M. Dunn. "Intuitive Semantics for First-Degree Entailments and 'Coupled Trees'." In: *Philosophical Studies* 29.3 (1976), pp. 149–168. DOI: 10.1007/bf00373152.

[118] C. Engel. "Inconsistency in the Law: In Search of a Balanced Norm." In: *SSRN Electronic Journal* (2004). ISSN: 1556-5068. DOI: 10.2139/ssrn.628387. URL: http://dx.doi.org/10.2139/ssrn.628387.

[119] A. FARQUHAR, R. FIKES, and J. RICE. "The Ontolingua Server: a tool for collaborative ontology construction." In: *International Journal of Human-Computer Studies* 46.6 (1997), pp. 707–727. ISSN: 1071-5819. DOI: https://doi.org/10.1006/ijhc.1996.0121. URL: https://www.sciencedirect.com/science/article/pii/S1071581996901214.

[120] D. Fensel et al. "OIL: an ontology infrastructure for the Semantic Web." In: *IEEE Intelligent Systems* 16.2 (2001), pp. 38–45. DOI: 10.1109/5254.920598.

[121] M. C. Fitting and R. L. Mendelsohn. *First-Order Modal Logic*. Dordrecht, Netherland: Kluwer Academic Publishers, 1998.

[122] E. Franconi, Y. A. Ibáñez-García, and İ. Seylan. "Query Answering with DBoxes is Hard." In: *Electronic Notes in Theoretical Computer Science* 278 (2011). Proceedings of the 7th Workshop on Methods for Modalities (M4M'2011) and the 4th Workshop on Logical Aspects of Multi-Agent Systems (LAMAS'2011), pp. 71–84. ISSN: 1571-0661. DOI: https://doi.org/10.1016/j.entcs.2011.10.007. URL: https://www.sciencedirect.com/science/article/pii/S1571066111001356.

[123] M. Frede. *Essays in Ancient Philosophy*. NED - New edition. University of Minnesota Press, 1987. ISBN: 9780816612741. URL: http://www.jstor.org/stable/10.5749/j.cttttpfd (visited on 04/04/2023).

[124] G. Frege. "Begriffsschrift." In: *From Frege to Gödel*. Ed. by J. V. Heijenoort. Cambridge: Harvard University Press, 1967, pp. 1–83.

[125] F. Freitas, H. Stuckenschmidt, and N. F. Noy. “Guest editor’s introduction: Ontology issues and applications.” In: *Journal of the Brazilian Computer Society* 11.2 (Nov. 2005), pp. 5–16. ISSN: 0104-6500. DOI: 10.1007/BF03192372. URL: https://doi.org/10.1007/BF03192372.

[126] S. French and K. McKenzie. “Rethinking Outside the Toolbox : Reflecting Again on the Relationship Between Philosophy of Science and Metaphysics.” In: *Metaphysics in Contemporary Physics*. Ed. by T. Bigaj and C. Wu?thrich. Brill | Rodopi, 2015.

[127] D. M. Gabbay. “Fibred Semantics and the Weaving of Logics Part 1: Modal and Intuitionistic Logics.” In: *Journal of Symbolic Logic* 61.4 (1996), pp. 1057–1120. DOI: 10.2307/2275807.

[128] D. M. Gabbay. “An Overview of Fibred Semantics and the Combination of Logics.” In: *Frontiers of Combining Systems: First International Workshop, Munich, March 1996*. Ed. by F. Baader and K. U. Schulz. Dordrecht: Springer Netherlands, 1996, pp. 1–55. ISBN: 978-94-009-0349-4. DOI: 10.1007/978-94-009-0349-4_1. URL: https://doi.org/10.1007/978-94-009-0349-4_1.

[129] A. Gangemi, D. Pisanelli, and G. Steve. “Ontology integration: Experiences with medical terminologies.” In: *Formal ontology in information systems*. Vol. 46. IOS Press Amsterdam. 1998, pp. 98–94.

[130] A. Gangemi et al. “Understanding top-level ontological distinctions.” In: *OIS@ IJCAI* 47 (2001).

[131] M. Gaur and A. Sheth. “Building trustworthy NeuroSymbolic AI Systems: Consistency, reliability, explainability, and safety.” In: *AI Mag.* 45.1 (Feb. 2024), pp. 139–155. ISSN: 0738-4602. DOI: 10.1002/aaai.12149. URL: https://doi.org/10.1002/aaai.12149.

[132] M. Genesereth. “Knowledge interchange format.” In: *Proceedings of the Conference on the Principles of Knowledge Representation and Reasoning*. Morgan Kaufmann Publishers. 1992.

[133] M. R. Genesereth and N. J. Nilsson. “CHAPTER 9 - Knowledge and Belief.” In: *Logical Foundations of Artificial Intelligence*. Ed. by M. R. Genesereth and N. J. Nilsson. San Francisco (CA): Morgan Kaufmann, 1987, pp. 207–238. ISBN: 978-0-934613-31-6. DOI: https://doi.org/10.1016/B978-0-934613-31-6.50015-X. URL: https://www.sciencedirect.com/science/article/pii/B978093461331650015X.

[134] J. H. Gennari et al. “The evolution of Protégé: an environment for knowledge-based systems development.” In: *International Journal of Human-computer studies* 58.1 (2003), pp. 89–123.

[135] J.-Y. Girard. “Linear logic.” In: *Theoretical Computer Science* 50.1 (1987), pp. 1–101. ISSN: 0304-3975. DOI: https://doi.org/10.1016/0304-3975(87)90045-4. URL: https://www.sciencedirect.com/science/article/pii/0304397587900454.

[136] J. Girard. “On the Unity of Logic.” In: *Annals of Pure and Applied Logic* 59.3 (1993), pp. 201–217. DOI: 10.1016/0168-0072(93)90093-s.

[137] J. Goguen and G. Roşu. “Institution morphisms.” In: *Formal aspects of computing* 13 (2002), pp. 274–307.

[138] J. A. Goguen. *Theorem Proving and Algebra*. 2021. arXiv: 2101.02690 [cs.LO].

[139] J. A. Goguen and R. M. Burstall. "Institutions: Abstract Model Theory for Specification and Programming." In: *J. ACM* 39.1 (Jan. 1992), pp. 95–146. ISSN: 0004-5411. DOI: 10.1145/147508.147524. URL: https://doi.org/10.1145/147508.147524.

[140] R. Goldblatt. *Topoi: the categorial analysis of logic*. Elsevier, 2014.

[141] A. Gómez-Pérez and O. Corcho. "Ontology languages for the semantic web." In: *IEEE Intelligent systems* 17.1 (2002), pp. 54–60.

[142] T. F. Gordon. "Constructing legal arguments with rules in the legal knowledge interchange format (LKIF)." In: *Computable models of the law: Languages, dialogues, games, ontologies*. Springer, 2008, pp. 162–184.

[143] B. C. Grau, B. Parsia, and E. Sirin. "Combining OWL ontologies using E-connections." In: *Journal of web semantics* 4.1 (2006), pp. 40–59.

[144] B. C. Grau et al. "Automatic partitioning of OWL ontologies using e-connections." In: *Description Logics* 147 (2005).

[145] R. Greiner and D. B. Lenat. "A Representation Language Language." In: *Proceedings of the First AAAI Conference on Artificial Intelligence*. AAAI'80. Stanford, California: AAAI Press, 1980, pp. 165–169.

[146] P. Grenon. "Chapter 3: A Primer on Knowledge Representation and Ontological Engineering." In: *Applied Ontology: An Introduction*. Ed. by K. Munn and B. Smith. Ontos, 2008, pp. 57–82.

[147] P. Grenon and B. Smith. "SNAP and SPAN: Prolegomenon to Geodynamic Ontology." In: 2003.

[148] P. Grenon and B. Smith. "SNAP and SPAN: Towards Dynamic Spatial Ontology." In: *Spatial Cognition & Computation* 4.1 (2004), pp. 69–104. DOI: 10.1207/s15427633scc0401_5. eprint: https://doi.org/10.1207/s15427633scc0401_5. URL: https://doi.org/10.1207/s15427633scc0401_5.

[149] T. R. Gruber. "A translation approach to portable ontology specifications." In: *Knowledge Acquisition* 5.2 (1993), pp. 199–220. ISSN: 1042-8143. DOI: https://doi.org/10.1006/knac.1993.1008. URL: https://www.sciencedirect.com/science/article/pii/S1042814383710083.

[150] T. R. Gruber. "Toward principles for the design of ontologies used for knowledge sharing?" In: *International Journal of Human-Computer Studies* 43.5 (1995), pp. 907–928. ISSN: 1071-5819. DOI: https://doi.org/10.1006/ijhc.1995.1081. URL: https://www.sciencedirect.com/science/article/pii/S1071581985710816.

[151] E. S. Gruen. *Rethinking the Other in Antiquity*. Princeton: Princeton University Press, 2010. ISBN: 9781400836550. DOI: doi:10.1515/9781400836550. URL: https://doi.org/10.1515/9781400836550.

[152] M. Grüninger et al. "TUpper: A Top Level Ontology within Standards1." In: *Appl. Ontol.* 17.1 (Jan. 2022), pp. 143–165. ISSN: 1570-5838. DOI: 10.3233/AO-220263. URL: https://doi.org/10.3233/AO-220263.

[153] N. Guarino. "21 Years of Applied Ontology." Invited presentation at Carnegie Mellon University. 2014. URL: https://www.slideshare.net/NicolaGuarino1/21-years-of-applied-ontology.

[154] N. Guarino. “Bfo and Dolce: So Far, so Close?” In: *Cosmos + Taxis* 4.4 (2017), pp. 10–18.

[155] N. Guarino. *Formal ontology in information systems: Proceedings of the first international conference (FOIS’98), June 6-8, Trento, Italy*. Vol. 46. IOS press, 1998.

[156] N. Guarino. “Formal ontology, conceptual analysis and knowledge representation.” In: *International Journal of Human-Computer Studies* 43.5 (1995), pp. 625–640. ISSN: 1071-5819. DOI: `https://doi.org/10.1006/ijhc.1995.1066`. URL: `https://www.sciencedirect.com/science/article/pii/S107158198571066X`.

[157] N. Guarino. “Semantic matching: Formal ontological distinctions for information organization, extraction, and integration.” In: *Information Extraction A Multidisciplinary Approach to an Emerging Information Technology*. Ed. by M. T. Pazienza. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 139–170. ISBN: 978-3-540-69548-6.

[158] N. Guarino. “Understanding, Building and Using Ontologies.” In: *Int. J. Hum.-Comput. Stud.* 46.2–3 (Mar. 1997), pp. 293–310. ISSN: 1071-5819. DOI: `10.1006/ijhc.1996.0091`. URL: `https://doi.org/10.1006/ijhc.1996.0091`.

[159] N. Guarino and M. A. Musen. “Applied Ontology: Focusing on Content.” In: *Applied ontology* 1.1 (2005), pp. 1–5.

[160] N. Guarino, D. Oberle, and S. Staab. “What Is an Ontology?” In: *Handbook on Ontologies*. Ed. by S. Staab and R. Studer. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 1–17. ISBN: 978-3-540-92673-3. DOI: `10.1007/978-3-540-92673-3_0`. URL: `https://doi.org/10.1007/978-3-540-92673-3_0`.

[161] N. Guarino, D. Porello, and G. Guizzardi. “On Weak Truthmaking.” In: *Proceedings of the Joint Ontology Workshops 2019*. Ed. by A. Barton, S. Seppälä, and D. Porello. CEUR Workshop Proceedings, 2019.

[162] N. Guarino, S. Uniti, and P. Giaretta. “Ontologies and Knowledge Bases. Towards a Terminological Clarification.” In: 1995.

[163] N. Guarino and C. Welty. “Ontological analysis of taxonomic relationships.” In: *International Conference on Conceptual Modeling*. Springer. 2000, pp. 210–224.

[164] G. Guizzardi. “On Ontology, ontologies, Conceptualizations, Modeling Languages, and (Meta)Models.” In: *International Baltic Conference on Databases and Information Systems*. 2007.

[165] G. Guizzardi. “Ontological Foundations for Structural Conceptual Models.” PhD thesis. Jan. 2005.

[166] G. Guizzardi et al. “Towards ontological foundations for conceptual modeling: The unified foundational ontology (UFO) story.” In: *Applied ontology* 10.3-4 (2015), pp. 259–271.

[167] G. Guizzardi et al. “UFO: Unified foundational ontology.” In: *Applied ontology* 17.1 (2022), pp. 167–210.

[168] V. Haarslev and R. Möller. “Consistency testing: The RACE experience.” In: *International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*. Springer. 2000, pp. 57–61.

[169] F. Harmelen, V. Lifschitz, and B. Porter. “The Handbook of Knowledge Representation.” In: *Elsevier Science San Diego, USA* (Jan. 2007), p. 1034.

[170] F. van Harmelen et al. *Handbook of Knowledge Representation*. San Diego, CA, USA: Elsevier Science, 2007. ISBN: 0444522115.

[171] J. Hartmann, R. Palma, and A. Gómez-Pérez. "Ontology Repositories." In: Springer, 2009, pp. 551–571. URL: `https://EconPapers.repec.org/RePEc:spr:ihichp:978-3-540-92673-3_25`.

[172] P. J. Hayes. *Naive Physics I: Ontology for Liquids*. Tech. rep. 35. Working Papers. Fondazione Dalle Molle, 1978.

[173] S. Hedman. *A First Course in Logic: An Introduction to Model Theory, Proof Theory, Computability, and Complexity*. New York: Oxford University Press, 2004.

[174] M. Heidegger. "Preface." In: *Heidegger: Through Phenomenology to Thought*. Ed. by W. J. Richardson. Fordham University Press, 1963.

[175] J. V. Heijenoort. "Logic as Calculus and Logic as Language." In: *Synthese* 17.1 (1967), pp. 324–330. DOI: `10.1007/bf00485036`.

[176] P. Hitzler et al. "Dolce+ D&S Ultralite and its main ontology design patterns." In: *Ontology engineering with ontology design patterns: foundations and applications* 25 (2016), p. 81.

[177] P. Hitzler. "A review of the semantic web field." In: *Communications of the ACM* 64.2 (2021), pp. 76–83.

[178] P. Hitzler et al. "OWL 2 web ontology language primer." In: *W3C recommendation* 27.1 (2009), p. 123.

[179] R. Hoekstra. "Ontology Representation: Design Patterns and Ontologies that Make Sense." In: *Proceedings of the 2009 Conference on Ontology Representation: Design Patterns and Ontologies That Make Sense*. NLD: IOS Press, 2009, pp. 1–236. ISBN: 9781607500131.

[180] L. R. Horn. "7Negation and Opposition: Contradiction and contrariety in logic and language." In: *The Oxford Handbook of Negation*. Oxford University Press, Mar. 2020. ISBN: 9780198830528. DOI: `10.1093/oxfordhb/9780198830528.013.1`. eprint: `https://academic.oup.com/book/0/chapter/352589885/chapter-ag-pdf/48439735/book\_41364\_section\_352589885.ag.pdf`. URL: `https://doi.org/10.1093/oxfordhb/9780198830528.013.1`.

[181] L. R. Horn. *A Natural History of Negation*. University of Chicago Press, 1989.

[182] M. Horridge and S. Bechhofer. "The OWL API: A Java API for OWL ontologies." In: *Semant. Web* 2.1 (Jan. 2011), pp. 11–21. ISSN: 1570-0844.

[183] I. Horrocks. "OWL: A Description Logic Based Ontology Language." In: *Principles and Practice of Constraint Programming - CP 2005*. Ed. by P. van Beek. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 5–8. ISBN: 978-3-540-32050-0.

[184] I. Horrocks. "Using an expressive description logic: FaCT or fiction?" In: *KR* 98 (1998), pp. 636–645.

[185] I. Horrocks, O. Kutz, and U. Sattler. "The Even More Irresistible SROIQ." In: *Kr* 6 (2006), pp. 57–67.

[186] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. “From SHIQ and RDF to OWL: the making of a Web Ontology Language.” In: *Journal of Web Semantics* 1.1 (2003), pp. 7–26. ISSN: 1570-8268. DOI: https://doi.org/10.1016/j.websem.2003.07.001. URL: https://www.sciencedirect.com/science/article/pii/S1570826803000027.

[187] E. Husserl. *Logical Investigations Volume 1*. Routledge, 2001.

[188] E. Husserl. *Logical Investigations Volume 2*. Routledge, 2001.

[189] J. Hyland. “Proof theory in the abstract.” In: *Annals of Pure and Applied Logic* 114.1 (2002). Troelstra Festschrift, pp. 43–78. ISSN: 0168-0072. DOI: https://doi.org/10.1016/S0168-0072(01)00075-6. URL: https://www.sciencedirect.com/science/article/pii/S0168007201000756.

[190] P. V. Inwagen. “Meta-Ontology.” In: *Erkenntnis (1975-)* 48.2/3 (1998), pp. 233–250. ISSN: 01650106, 15728420. URL: http://www.jstor.org/stable/20012842 (visited on 03/17/2023).

[191] *Information technology — Top-level ontologies (TLO) — Part 2: Basic Formal Ontology (BFO)*. Standard. Geneva, CH: International Organization for Standardization, Nov. 2021.

[192] *Information technology — Common Logic (CL) — A framework for a family of logic-based languages*. Standard. Geneva, CH: International Organization for Standardization, Nov. 2018.

[193] M. Jackermeier, J. Chen, and I. Horrocks. “Dual box embeddings for the description logic EL++.” In: *Proceedings of the ACM on Web Conference 2024*. 2024, pp. 2250–2258.

[194] R. Jackson et al. “OBO Foundry in 2021: operationalizing open data principles to evaluate ontologies.” In: *Database* 2021 (Oct. 2021), baab069. ISSN: 1758-0463. DOI: 10.1093/database/baab069. eprint: https://academic.oup.com/database/article-pdf/doi/10.1093/database/baab069/40854912/baab069.pdf. URL: https://doi.org/10.1093/database/baab069.

[195] B. Jacobs. *Categorical logic and type theory*. Elsevier, 1999.

[196] D. Jacquette. *Ontology*. Routledge, 2002.

[197] U. Jaimini, C. Henson, and A. Sheth. “Causal Neurosymbolic AI: A Synergy Between Causality and Neurosymbolic Methods.” In: *IEEE Intelligent Systems* 39.03 (May 2024), pp. 13–19. ISSN: 1941-1294. DOI: 10.1109/MIS.2024.3395936.

[198] L. Jansen. “Chapter 8: Categories: The Top-Level Ontology.” In: *An Introduction*. Ed. by K. Munn and B. Smith. Berlin, Boston: De Gruyter, 2008, pp. 173–196. ISBN: 9783110324860. DOI: doi:10.1515/9783110324860.173. URL: https://doi.org/10.1515/9783110324860.173.

[199] M. Jarke et al. “KBMS requirements of knowledge-based systems.” In: *Foundations of Knowledge Base Management: Contributions from Logic, Databases, and Artificial Intelligence Applications* (1989), pp. 381–394.

[200] E. Jiménez-Ruiz et al. “Ontology integration using mappings: Towards getting the right logical consequences.” In: *The Semantic Web: Research and Applications: 6th European Semantic Web Conference, ESWC 2009 Heraklion, Crete, Greece, May 31–June 4, 2009 Proceedings 6*. Springer. 2009, pp. 173–187.

[201] C. H. Kahn. "The Thesis of Parmenides." In: *The Review of Metaphysics* 22.4 (1969), pp. 700–724. ISSN: 00346632. URL: `http://www.jstor.org/stable/20124945` (visited on 03/23/2023).

[202] N. Kamide. "Paraconsistent description logics revisited." In: *23rd International Workshop on Description Logics DL2010*. 2010, p. 197.

[203] I. Kant. "On the ground of the distinction of all objects in general into phenomena and noumena (as in the second edition)." In: *Critique of Pure Reason*. Ed. by P. Guyer and A. W. Wood. The Cambridge Edition of the Works of Immanuel Kant. Cambridge University Press, 1998. DOI: `10.1017/CBO9780511804649.019`.

[204] M. H. Karray et al. "ROMAIN: Towards a BFO compliant reference ontology for industrial maintenance." In: *Applied Ontology* 14.2 (2019), pp. 155–177.

[205] H. Kim. "Locke on Substance." In: *The Lockean Mind*. Ed. by J. Gordon-Roth and S. Weinberg. New York and London: Routledge, 2021, pp. 226–236.

[206] T. K. Kissel. "Logical Pluralism From a Pragmatic Perspective." In: *Australasian Journal of Philosophy* 96.3 (2018), pp. 578–591. DOI: `10.1080/00048402.2017.1399151`.

[207] P. Klinov. "Pronto: A non-monotonic probabilistic description logic reasoner." In: *European Semantic Web Conference*. Springer. 2008, pp. 822–826.

[208] S. A. Kripke. "The Undecidability of Monadic Modal Quantification Theory." In: *Mathematical Logic Quarterly* 8.2 (1962), pp. 113–116. DOI: `https://doi.org/10.1002/malq.19620080204`. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/malq.19620080204`. URL: `https://onlinelibrary.wiley.com/doi/abs/10.1002/malq.19620080204`.

[209] B. Kulvatunyou et al. "The Industrial Ontologies Foundry (IOF) Core Ontology." en. In: Formal Ontologies Meet Industry (FOMI) 2022, Tarbes, FR, Sept. 2022. URL: `https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=935068`.

[210] G. Küng. "The phenomenological reduction as epoche and as explication." In: *The Monist* (1975), pp. 63–80.

[211] O. Kutz. "$\mathcal{E}$-connections and Logics of Distance." PhD thesis. University of Liverpool, 2004.

[212] O. Kutz and T. Mossakowski. "A Modular Consistency Proof for DOLCE." In: *Proceedings of the AAAI Conference on Artificial Intelligence* 25.1 (Aug. 2011), pp. 227–234. DOI: `10.1609/aaai.v25i1.7860`. URL: `https://ojs.aaai.org/index.php/AAAI/article/view/7860`.

[213] O. Kutz et al. "E-connections of abstract description systems." In: *Artificial intelligence* 156.1 (2004), pp. 1–73.

[214] J. Lambek. "The Mathematics of Sentence Structure." In: *The American Mathematical Monthly* 65.3 (1958), pp. 154–170. DOI: `10.1080/00029890.1958.11989160`. eprint: `https://doi.org/10.1080/00029890.1958.11989160`. URL: `https://doi.org/10.1080/00029890.1958.11989160`.

[215] P. J. Landin. "The Mechanical Evaluation of Expressions." In: *The Computer Journal* 6.4 (Jan. 1964), pp. 308–320. ISSN: 0010-4620. DOI: `10.1093/comjnl/6.4.308`. eprint: `https://academic.oup.com/comjnl/article-pdf/6/4/308/1067901/6-4-308.pdf`. URL: `https://doi.org/10.1093/comjnl/6.4.308`.

[216] C. Lange et al. *The Distributed Ontology Language (DOL): Use Cases, Syntax, and Extensibility*. 2012. arXiv: 1208.0293 [cs.AI].

[217] O. Lassila, J. Hendler, and T. Berners-Lee. "The semantic web." In: *Scientific American* 284.5 (2001), pp. 34–43.

[218] O. Laurent. "Étude de la polarisation en logique." Theses. Université de la Méditerranée - Aix-Marseille II, Mar. 2002. URL: https://theses.hal.science/tel-00007884.

[219] O. Laurent. "Polarized proof-nets and $\lambda\mu$-calculus." In: *Theoretical Computer Science* 290.1 (2003), pp. 161–188.

[220] O. Laurent. "Syntax vs. semantics: a polarized approach." In: *Theoretical Computer Science* 343.1-2 (2005), pp. 177–206.

[221] G. W. Leibniz. "The Monadology." In: *Early Modern Philosophy: Essential Readings with Commentary*. Ed. by A. Martinich, F. Allhoff, and A. Vaidya. Blackwell, 1989 [1714].

[222] W. Leinfellner. *Language and ontology : proceedings of the sixth International Wittgenstein Symposium 23rd to 30th August 1981, Kirchberg am Wechsel, Austria / editors, Werner Leinfellner, Eric Kraemer, and Jeffrey Schank.* eng;ger. Schriftenreihe der Wittgenstein-Gesellschaft ; v. 8. Vienna: Holder-Pichler-Tempsky, 1982. ISBN: 3209004226.

[223] H. Leitgeb and A. Carus. "Rudolf Carnap." In: *The Stanford Encyclopedia of Philosophy*. Ed. by E. N. Zalta and U. Nodelman. Fall 2022. Metaphysics Research Lab, Stanford University, 2022.

[224] D. Lembo et al. "Inconsistency-Tolerant Semantics for Description Logics." In: *Web Reasoning and Rule Systems*. Ed. by P. Hitzler and T. Lukasiewicz. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 103–117. ISBN: 978-3-642-15918-3.

[225] D. Lenat, M. Prakash, and M. Shepherd. "CYC: Using Common Sense Knowledge to Overcome Brittleness and Knowledge Acquistion Bottlenecks." In: *AI Mag.* 6.4 (Jan. 1986), pp. 65–85. ISSN: 0738-4602.

[226] D. B. Lenat and R. V. Guha. *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project*. 1st. USA: Addison-Wesley Longman Publishing Co., Inc., 1989. ISBN: 0201517523.

[227] D. B. Lenat and R. V. Guha. "The Evolution of CycL, the Cyc Representation Language." In: *SIGART Bull.* 2.3 (June 1991), pp. 84–87. ISSN: 0163-5719. DOI: 10.1145/122296.122308. URL: https://doi.org/10.1145/122296.122308.

[228] H. J. Levesque and R. J. Brachman. "Expressiveness and tractability in knowledge representation and reasoning1." In: *Computational Intelligence* 3.1 (1987), pp. 78–93. DOI: https://doi.org/10.1111/j.1467-8640.1987.tb00176.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8640.1987.tb00176.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8640.1987.tb00176.x.

[229] C. Liang and D. Miller. "Focusing and polarization in linear, intuitionistic, and classical logics." In: *Theoretical Computer Science* 410.46 (2009), pp. 4747–4768.

[230] L. Libkin. "Other Applications of Finite Model Theory." In: *Elements of Finite Model Theory*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 275–289. ISBN: 978-3-662-07003-1. DOI: 10.1007/978-3-662-07003-1_14. URL: https://doi.org/10.1007/978-3-662-07003-1_14.

[231] F. Loebe, P. Burek, and H. Herre. "GFO: The General Formal Ontology." In: *Applied Ontology* 17 (Feb. 2022), pp. 1–36. DOI: 10.3233/AO-220264.

[232] A. Lopes et al. "How to Classify Domain Entities Into Top-Level Ontology Concepts Using Large Language Models." In: *Applied ontology* (2024), pp. 1–29. DOI: 10.3233/ao-240032.

[233] E. J. Lowe. *The Four-Category Ontology: A Metaphysical Foundation for Natural Science*. Oxford University Press, Dec. 2005. ISBN: 9780199254392. DOI: 10.1093/0199254397.001.0001. URL: https://doi.org/10.1093/0199254397.001.0001.

[234] D. Lucanu, Y. F. Li, and J. S. Dong. "Semantic Web Languages – Towards an Institutional Perspective." In: *Algebra, Meaning, and Computation: Essays dedicated to Joseph A. Goguen on the Occasion of His 65th Birthday*. Ed. by K. Futatsugi, J.-P. Jouannaud, and J. Meseguer. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 99–123. ISBN: 978-3-540-35464-2. DOI: 10.1007/11780274_6. URL: https://doi.org/10.1007/11780274_6.

[235] D. Lücke. "Carnap, Goguen, and the Hyperontologies: Logical Pluralism and Heterogeneous Structuring in Ontology Design." In: *Logica Universalis* 4.2 (2010), pp. 255–333. DOI: 10.1007/s11787-010-0020-3.

[236] Y. Ma, P. Hitzler, and Z. Lin. "Algorithms for Paraconsistent Reasoning with OWL." In: *Proceedings of the 4th European Conference on The Semantic Web: Research and Applications*. ESWC '07. Innsbruck, Austria: Springer-Verlag, 2007, pp. 399–413. ISBN: 9783540726661. DOI: 10.1007/978-3-540-72667-8_29. URL: https://doi.org/10.1007/978-3-540-72667-8_29.

[237] Y. Ma, P. Hitzler, and Z. Lin. "Paraconsistent Reasoning for Expressive and Tractable Description Logics." In: *Description Logics* 353 (2008).

[238] S. Maclane. *Categories for the Working Mathematician*. Springer, 1971.

[239] A. Maedche and S. Staab. "The Karlsruhe Ontology and Semantic Web Meta Project." In: *Knstliche Intelligenz* 3 (2003), pp. 27–30.

[240] L. Magee. "9 - Upper-level ontologies." In: *Towards a Semantic Web*. Ed. by B. Cope, M. Kalantzis, and L. Magee. Chandos Publishing, 2011, pp. 235–287. ISBN: 978-1-84334-601-2. DOI: https://doi.org/10.1016/B978-1-84334-601-2.50009-X. URL: https://www.sciencedirect.com/science/article/pii/B978184334601250009X.

[241] F. Marcacci. "Argumentation and counterfactual reasoning in Parmenides and Melissus." In: *Revista Archai* 30 (May 2020), e03004. DOI: 10.14195/1984-249X_30_4. URL: https://impactum-journals.uc.pt/archai/article/view/1984-249X_30_4.

[242] S. Marcelino and C. Caleiro. "Decidability and Complexity of Fibred Logics Without Shared Connectives." In: *Logic Journal of the IGPL* 24.5 (2016). DOI: 10.1093/jigpal/jzw033.

[243] S. Marcelino and C. Caleiro. "On the characterization of fibred logics, with applications to conservativity and finite-valuedness." In: *Journal of Logic and Computation* 27.7 (Aug. 2016), pp. 2063–2088. ISSN: 0955-792X. DOI: 10.1093/logcom/exw023. eprint: https://academic.oup.com/logcom/article-pdf/27/7/2063/21322525/exw023.pdf. URL: https://doi.org/10.1093/logcom/exw023.

[244] S. Marcelino, C. Caleiro, and P. Baltazar. "Deciding Theoremhood in Fibred Logics Without Shared Connectives." In: *The Road to Universal Logic: Festschrift for the 50th Birthday of Jean-Yves Béziau Volume II*. Ed. by A. Koslow and A. Buchsbaum. Cham: Springer International Publishing, 2015, pp. 387–406. ISBN: 978-3-319-15368-1. DOI: 10.1007/978-3-319-15368-1_18. URL: https://doi.org/10.1007/978-3-319-15368-1_18.

[245] J. Marcos. "Possible-translations semantics." In: *Proceedings of the Workshop on Combination of Logics: Theory and applications (CombLog'04), held in Lisbon, PT*. 2004, pp. 119–128.

[246] G. Marcus. *No AGI (and no Trustworthy AI) without Neurosymbolic AI*. https://ontologforum.com/index.php/ConferenceCall_2024_02_28. Accessed: 2024–08-11. 2024.

[247] S. Marin et al. "A Pure View of Ecumenical Modalities." In: *Logic, Language, Information, and Computation: 27th International Workshop, Wollic 2021, Virtual Event, October 5?8, 2021, Proceedings*. Ed. by A. Silva, R. Wassermann, and R. de Queiroz. Springer Verlag, 2021, pp. 388–407.

[248] J. Marques-Silva. "Logic-based explainability in machine learning." In: *Reasoning Web. Causality, Explanations and Declarative Knowledge: 18th International Summer School 2022, Berlin, Germany, September 27–30, 2022, Tutorial Lectures*. Springer, 2023, pp. 24–104.

[249] V. Mascardi, V. Cordì, P. Rosso, et al. "A Comparison of Upper Ontologies." In: *Woa*. Vol. 2007. 2007, pp. 55–64.

[250] C. Masolo et al. "WonderWeb Deliverable D18 Ontology Library." In: 2003.

[251] J. McCarthy and P. Hayes. "Some Philosophical Problems From the Standpoint of Artificial Intelligence." In: *Machine Intelligence 4*. Ed. by B. Meltzer and D. Michie. Edinburgh University Press, 1969, pp. 463–502.

[252] S. A. Meirelles. "Beyond monism and pluralism in logic." MA thesis. Universidade Estadual de Campinas, Instituto de Filosofia e Ciências Humanas, 2023.

[253] S. Menn. "The Stoic Theory of Categories." In: *Oxford Studies in Ancient Philosophy* 17 (1999), pp. 215–47.

[254] J. Meseguer. "General logics." In: *Studies in Logic and the Foundations of Mathematics*. Vol. 129. Elsevier, 1989, pp. 275–329.

[255] A. C. Michalos. *The Popper-Carnap Controversy*. The Hague, M. Nijhoff, 1971.

[256] I. Mikenberg, N. C. A. D. Costa, and R. Chuaqui. "Pragmatic Truth and Approximation to Truth." In: *The Journal of Symbolic Logic* 51.1 (1986), pp. 201–221. ISSN: 00224812. URL: http://www.jstor.org/stable/2273956 (visited on 08/23/2023).

[257] M. Minsky. *A framework for representing knowledge*. 1974.

[258] R. Mizoguchi et al. "YAMATO: Yet-Another More Advanced Top-Level Ontology." In: *Appl. Ontol.* 17.1 (Jan. 2022), pp. 211–232. ISSN: 1570-5838. DOI: 10.3233/AO-210257. URL: https://doi.org/10.3233/AO-210257.

[259] B. Mohapatra et al. "EmELvar: A NeuroSymbolic Reasoner for the EL++ Description Logic." In: *SemREC@ ISWC*. 2021, pp. 44–51.

[260] T. Mossakowski. "Comorphism-Based Grothendieck Logics." In: *Mathematical Foundations of Computer Science 2002*. Ed. by K. Diks and W. Rytter. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 593–604. ISBN: 978-3-540-45687-2.

[261] T. Mossakowski. *Heterogeneous Specification and the Heterogeneous Tool Set Habilitation thesis*. 2005.

[262] T. Mossakowski. *Neuro-symbolic integration for ontology-based classification of structured objects*. https://ontologforum.com/index.php/ConferenceCall_2024_03_20. Accessed: 2024–08-11. 2024.

[263] T. Mossakowski. "The Distributed Ontology, Model and Specification Language – DOL." In: *WADT 2016*. Ed. by M. Roggenbach. Vol. 10644. LNCS. Springer, 2017, pp. 5–10. DOI: 10.1007/978-3-319-72044-9_2. URL: https://link.springer.com/chapter/10.1007/978-3-319-72044-9_2.

[264] T. Mossakowski, R. Diaconescu, and A. Tarlecki. "What is a Logic Translation?" In: *Logica Universalis* 3.1 (2009), pp. 95–124. DOI: 10.1007/s11787-009-0005-2.

[265] T. Mossakowski, C. Maeder, and K. Lüttich. "The heterogeneous tool set, Hets." In: *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer. 2007, pp. 519–522.

[266] T. Mossakowski et al. "What is a Logic?" In: *Logica Universalis*. Ed. by J.-Y. Beziau. Basel: Birkhäuser Basel, 2007, pp. 111–133. ISBN: 978-3-7643-8354-1.

[267] C. Moustakas. *Phenomenological research methods*. Sage publications, 1994.

[268] A.-R. Muljarto et al. "Ontology-Based Model for Food Transformation Processes - Application to Winemaking." In: *Metadata and Semantics Research*. Ed. by S. Closs et al. Cham: Springer International Publishing, 2014, pp. 329–343. ISBN: 978-3-319-13674-5.

[269] M. A. Musen et al. "The national center for biomedical ontology." In: *Journal of the American Medical Informatics Association* 19.2 (2012), pp. 190–195.

[270] M. A. Musen. "The protégé project: a look back and a look forward." In: *AI matters* 1 4 (2015), pp. 4–12. URL: https://api.semanticscholar.org/CorpusID:13208034.

[271] P. P. Negri et al. "Towards an Ontology of Goal-Oriented Requirements." In: *CIbSE* 94 (2017), pp. 469–482.

[272] D. Nelson. "Constructible Falsity." In: *The Journal of Symbolic Logic* 14.1 (1949), pp. 16–26. ISSN: 00224812. URL: http://www.jstor.org/stable/2268973 (visited on 06/02/2024).

[273] F. Neuhaus. "On the Definition of'Ontology'." In: *JOWO*. 2017.

[274] F. Neuhaus. "Ontologies in the era of large language models–a perspective." In: *Applied ontology* 18.4 (2023), pp. 399–407.

[275] F. Neuhaus and J. Hastings. "Ontology Development is Consensus Creation, Not (Merely) Representation." In: *Applied ontology* 17.4 (2022), pp. 495–513. DOI: 10.3233/ao-220273.

[276] A. Newell, J. C. Shaw, and H. A. Simon. "Report on a general problem-solving program." In: *Proceedings of the International Conference on Information Processing.* 1959, pp. 256–264.

[277] N. Ngo, M. Ortiz, and M. Šimkus. "Closed predicates in description logics: results on combined complexity." In: *Proceedings of the Fifteenth International Conference on Principles of Knowledge Representation and Reasoning.* KR'16. Cape Town, South Africa: AAAI Press, 2016, pp. 237–246.

[278] I. Niles and A. Pease. "Towards a standard upper ontology." In: *Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001.* 2001, pp. 2–9.

[279] S. Odintsov and H. Wansing. "Inconsistency-tolerant description logic. Part II: A tableau algorithm for CALCC." In: *Journal of Applied Logic* 6.3 (2008), pp. 343–360. ISSN: 1570-8683. DOI: https://doi.org/10.1016/j.jal.2007.06.001. URL: https://www.sciencedirect.com/science/article/pii/S1570868307000493.

[280] S. P. Odintsov and H. Wansing. "Inconsistency-tolerant Description Logic: Motivation and Basic Systems." In: *Trends in Logic: 50 Years of Studia Logica.* Ed. by V. F. Hendricks and J. Malinowski. Dordrecht: Springer Netherlands, 2003, pp. 301–335. ISBN: 978-94-017-3598-8. DOI: 10.1007/978-94-017-3598-8_11. URL: https://doi.org/10.1007/978-94-017-3598-8_11.

[281] C. Ogden and I. Richards. *The Meaning of Meaning: A Study of the Influence of Language Upon Thought and of the Science of Symbolism. Supplementary Essays by B. Malinowski and F.G. Crookshank.* Harvest book. Harcourt, 1923. URL: https://books.google.com.br/books?id=i3MIAQAAIAAJ.

[282] H. Omori and H. Wansing. "40 years of FDE: An Introductory Overview." In: *Studia Logica: An International Journal for Symbolic Logic* 105.6 (2017), pp. 1021–1049. ISSN: 00393215, 15728730. URL: http://www.jstor.org/stable/45103948 (visited on 07/10/2023).

[283] I. Osman, S. Ben Yahia, and G. Diallo. "Ontology Integration: Approaches and Challenging Issues." In: *Information Fusion* 71 (2021), pp. 38–63. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2021.01.007. URL: https://www.sciencedirect.com/science/article/pii/S1566253521000130.

[284] J. N. Otte, J. Beverley, and A. Ruttenberg. "Bfo: Basic Formal Ontology." In: *Applied ontology* 17.1 (2022), pp. 17–43. DOI: 10.3233/ao-220262.

[285] L. Palopoli, D. Saccà, and D. Ursino. "DLP: a description logic for extracting and managing complex terminological and structural properties from database schemes." In: *Information Systems* 24.5 (1999), pp. 401–424.

[286] B. Parsia and B. Cuenca Grau. "Generalized Link Properties for Expressive E-connections of Description Logics." In: (2005).

[287] C. Partridge et al. "A Survey of Top-Level Ontologies-to inform the ontological choices for a Foundation Data Model." In: (2020).

[288] P. Patel-Schneider. "System description: DLP." In: *Automated Deduction-CADE-17: 17th International Conference on Automated Deduction Pittsburgh, PA, USA, June 17-20, 2000. Proceedings 17*. Springer. 2000, pp. 297–301.

[289] P. F. Patel-Schneider. "A Four-Valued Semantics for Terminological Logics." In: *Artificial Intelligence* 38.3 (1989), pp. 319–351. DOI: 10.1016/0004-3702(89)90036-2.

[290] P. F. Patel-Schneider et al. "The CLASSIC knowledge representation system: guiding principles and implementation rationale." In: *SIGART Bull.* 2.3 (June 1991), pp. 108–113. ISSN: 0163-5719. DOI: 10.1145/122296.122313. URL: https://doi.org/10.1145/122296.122313.

[291] E. Patterson, O. Lynch, and J. Fairbanks. "Categorical Data Structures for Technical Computing." In: *Compositionality* 4 (5 Dec. 2022). ISSN: 2631-4444. DOI: 10.32408/compositionality-4-5. URL: https://doi.org/10.32408/compositionality-4-5.

[292] H. Paulheim and A. Gangemi. "Serving DBpedia with DOLCE – More than Just Adding a Cherry on Top." In: *The Semantic Web - ISWC 2015*. Ed. by M. Arenas et al. Cham: Springer International Publishing, 2015, pp. 180–196. ISBN: 978-3-319-25007-6.

[293] L. C. Paulson. "The foundation of a generic theorem prover." In: *Journal of Automated Reasoning* 5 (1989), pp. 363–397.

[294] A. Pease. "A Programmer's Text Editor for a Logical Theory: The SUMOjEdit Editor (System Description)." In: *Automated Reasoning: 10th International Joint Conference, IJCAR 2020, Paris, France, July 1–4, 2020, Proceedings, Part II*. Paris, France: Springer-Verlag, 2020, pp. 472–479. ISBN: 978-3-030-51053-4. DOI: 10.1007/978-3-030-51054-1_31. URL: https://doi.org/10.1007/978-3-030-51054-1_31.

[295] A. Pease. *Ontology: A Practical Guide*. Angwin, CA: Articulate Software Press, 2011. ISBN: 978-1-889455-10-5.

[296] A. Pease and S. Schulz. "Knowledge Engineering for Large Ontologies with Sigma KEE 3.0." In: *Automated Reasoning*. Ed. by S. Demri, D. Kapur, and C. Weidenbach. Cham: Springer International Publishing, 2014, pp. 519–525. ISBN: 978-3-319-08587-6.

[297] L. C. Pereira and E. Pimentel. "On an Ecumenical Natural Deduction with Stoup. Part I: The Propositional Case." In: *Perspectives on Deduction: Contemporary Studies in the Philosophy, History and Formal Theories of Deduction*. Ed. by A. Piccolomini d'Aragona. Cham: Springer International Publishing, 2024, pp. 139–169. ISBN: 978-3-031-51406-7. DOI: 10.1007/978-3-031-51406-7_7. URL: https://doi.org/10.1007/978-3-031-51406-7_7.

[298] E. Pimentel and L. C. Pereira. "A Tour on Ecumenical Systems." In: *Leibniz International Proceedings in Informatics, LIPIcs*. Vol. 270. Proceedings of the 10th Conference on Algebra and Coalgebra in Computer … 2023, p. 3.

[299] H. S. Pinto and J. P. Martins. "A methodology for ontology integration." In: *Proceedings of the 1st international conference on Knowledge capture*. 2001, pp. 131–138.

[300] R. Poli. "Descriptive, Formal and Formalized Ontologies." In: *Husserl's Logical Investigations Reconsidered*. Dordrecht: Springer Netherlands, 2003, pp. 183–210. ISBN: 978-94-017-0207-2. DOI: 10.1007/978-94-017-0207-2_12. URL: https://doi.org/10.1007/978-94-017-0207-2_12.

[301] R. Poli. "Husserl's Conception of Formal Ontology." In: *History and Philosophy of Logic* 14.1 (1993), pp. 1–14. DOI: 10.1080/01445349308837207.

[302] R. Poli and J. Seibt. *Theory and applications of ontology: Philosophical perspectives.* Springer, 2010.

[303] D. L. Poole and A. K. Mackworth. *Artificial Intelligence: Foundations of Computational Agents*. 2nd. USA: Cambridge University Press, 2017. ISBN: 110719539X.

[304] K. R. Popper. *The Logic of Scientific Discovery.* Ed. by H. P. Group. New York: Routledge, 1934.

[305] D. Porelo et al. *A First-Order Logic Formalization of the Unified Foundational Ontology.* Tech. rep. 35. Dec. 2021.

[306] D. Prawitz. "Classical versus intuitionistic logic." In: *Why is this a Proof? : Festschrift for Luiz Carlos Pereira.* Tributes 27. 2015, pp. 15–32. ISBN: 9781848901728.

[307] D. Prawitz. "Meaning and Proofs: On the Conflict Between Classical and Intuitionistic Logic." In: *Theoria* 43.1 (1977), pp. 2–40. DOI: 10.1111/j.1755-2567.1977.tb00776.x.

[308] G. A. Press. "The State of the Question in the Study of Plato." In: *The Southern Journal of Philosophy* 34.4 (1996), pp. 507–532. DOI: https://doi.org/10.1111/j.2041-6962.1996.tb00805.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.2041-6962.1996.tb00805.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.2041-6962.1996.tb00805.x.

[309] H. Price. "Metaphysics After Carnap : The Ghost Who Walks?" In: *Metametaphysics: New Essays on the Foundations of Ontology.* Ed. by D. Chalmers, D. Manley, and R. Wasserman. Oxford University Press, 2007, pp. 320–46.

[310] G. Priest. *An Introduction to Non-Classical Logic: From If to Is.* 2nd ed. Cambridge Introductions to Philosophy. Cambridge University Press, 2008. DOI: 10.1017/CBO9780511801174.

[311] G. Priest. *Towards non-being.* Oxford University Press, 2016.

[312] M. R. Quillian. "Semantic Networks." In: *Semantic Information Processing.* Ed. by M. L. Minsky. MIT Press, 1968.

[313] W. V. Quine. "On Carnap's Views on Ontology." In: *Philosophical Studies: An International Journal for Philosophy in the Analytic Tradition* 2.5 (1951), pp. 65–72. ISSN: 00318116, 15730883. URL: http://www.jstor.org/stable/4318118 (visited on 05/26/2024).

[314] W. V. Quine. "On What There Is." In: *The Review of Metaphysics* 2.5 (1948), pp. 21–38. ISSN: 00346632. URL: http://www.jstor.org/stable/20123117 (visited on 04/21/2023).

[315] W. V. O. Quine. "Two Dogmas of Empiricism." In: *Philosophical Review* 60.1 (1951), pp. 20–43. DOI: 10.2307/2266637.

[316] G. Rasch. *Probabilistic Models for Some Intelligence and Attainment Tests*. Studies in mathematical psychology. Danmarks Paedagogiske Institut, 1960. ISBN: 9780598554512. URL: https://books.google.com.br/books?id=aB9qLgEACAAJ.

[317] J. Rasga, A. Sernadas, and C. Sernadas. "Fibring as Biporting Subsumes Asymmetric Combinations." In: *Studia Logica: An International Journal for Symbolic Logic* 102.5 (2014), pp. 1041–1074. ISSN: 00393215, 15728730. URL: http://www.jstor.org/stable/43649674 (visited on 02/06/2024).

[318] J. Rasga, A. Sernadas, and C. Sernadas. "Importing Logics: Soundness and Completeness Preservation." In: *Studia Logica: An International Journal for Symbolic Logic* 101.1 (2013), pp. 117–155. ISSN: 00393215, 15728730. URL: http://www.jstor.org/stable/23488585 (visited on 02/06/2024).

[319] J. Rasga, A. Sernadas, and C. Sernadas. "Importing Logics." In: *Studia Logica* 100 (2012), pp. 545–581. URL: https://api.semanticscholar.org/CorpusID:16034063.

[320] J. RASGA, C. SERNADAS, and A. SERNADAS. "PRESERVATION OF ADMISSIBLE RULES WHEN COMBINING LOGICS." In: *The Review of Symbolic Logic* 9.4 (2016), pp. 641–663. DOI: 10.1017/S1755020316000241.

[321] W. Rautenberg. *A Concise Introduction to Mathematical Logic*. New York, NY: Springer, 2006.

[322] D. Rayside and G. Campbell. "An Aristotelian Understanding of Object-Oriented Programming." In: *ACM SIGPLAN Notices* 35 (Jan. 2001). DOI: 10.1145/353171.353194.

[323] D. Rayside and G. T. Campbell. "Aristotle and Object-Oriented Programming: Why Modern Students Need Traditional Logic." In: *Proceedings of the Thirty-First SIGCSE Technical Symposium on Computer Science Education*. SIGCSE '00. Austin, Texas, USA: Association for Computing Machinery, 2000, pp. 237–244. ISBN: 1581132131. DOI: 10.1145/330908.331862. URL: https://doi.org/10.1145/330908.331862.

[324] A. Richardson and D. Isaacson. "Carnap's Principle of Tolerance." In: *Proceedings of the Aristotelian Society, Supplementary Volumes* 68 (1994), pp. 67–83. ISSN: 03097013, 14678349. URL: http://www.jstor.org/stable/4107023.

[325] E. Riehl. *Category theory in context*. Courier Dover Publications, 2017.

[326] D. G. Robertson. "Stoic and Aristotelian Notions of Substance in Basil of Caesarea." In: *Vigiliae Christianae* 52.4 (1998), pp. 393–417. ISSN: 00426032. URL: http://www.jstor.org/stable/1584833 (visited on 04/07/2023).

[327] H. Robinson. "Substance." In: *The Stanford Encyclopedia of Philosophy*. Ed. by E. N. Zalta. Fall 2021. Metaphysics Research Lab, Stanford University, 2021.

[328] J. A. Robinson. "A Machine-Oriented Logic Based on the Resolution Principle." In: *J. ACM* 12.1 (Jan. 1965), pp. 23–41. ISSN: 0004-5411. DOI: 10.1145/321250.321253. URL: https://doi.org/10.1145/321250.321253.

[329] R. Routley. *Exploring Meinong's jungle and beyond: an investigation of noneism and the theory of items*. Australian National University Canberra, 1980.

[330] S. Rudolph. “Foundations of Description Logics.” In: *Reasoning Web. Semantic Technologies for the Web of Data: 7th International Summer School 2011, Galway, Ireland, August 23-27, 2011, Tutorial Lectures*. Ed. by A. Polleres et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 76–136. ISBN: 978-3-642-23032-5. DOI: 10.1007/978-3-642-23032-5_2. URL: https://doi.org/10.1007/978-3-642-23032-5_2.

[331] G. Russell. “Logical Pluralism.” In: *The Stanford Encyclopedia of Philosophy*. Ed. by E. N. Zalta. Summer 2021. Metaphysics Research Lab, Stanford University, 2021.

[332] G. K. Saha. “Web ontology language (OWL) and semantic web.” In: *Ubiquity* 2007.September (2007), pp. 1–1.

[333] D. Sannella. “The Common Framework Initiative for Algebraic Specification and Development of Software.” In: *Perspectives of System Informatics*. Ed. by D. Bjøner, M. Broy, and A. V. Zamulin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 1–9. ISBN: 978-3-540-46562-1.

[334] N. O. Santos et al. “O3PO: A domain ontology for offshore petroleum production plants.” In: *Expert Systems with Applications* 238 (2024), p. 122104. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2023.122104. URL: https://www.sciencedirect.com/science/article/pii/S0957417423026064.

[335] K. Sayre. *Plato's Late Ontology: A Riddle Resolved : with a New Introduction and the Essay, "Excess and Deficiency at Statesman 283C-285C"*. Parmenides Pub., 2005. ISBN: 9781930972094. URL: https://books.google.com.br/books?id=IrbWAAAAMAAJ.

[336] J. Schaffer. “On What Grounds What.” In: *Metametaphysics: New Essays on the Foundations of Ontology*. Ed. by D. Manley, D. J. Chalmers, and R. Wasserman. Oxford University Press, 2009, pp. 347–383.

[337] K. Schild. “A correspondence theory for terminological logics: preliminary report.” In: *Proceedings of the 12th International Joint Conference on Artificial Intelligence - Volume 1*. IJCAI'91. Sydney, New South Wales, Australia: Morgan Kaufmann Publishers Inc., 1991, pp. 466–471. ISBN: 1558601600.

[338] M. Schmidt-Schaubß. “Subsumption in KL-ONE is undecidable.” In: *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1989, pp. 421–431. ISBN: 1558600329.

[339] M. Schmidt-Schauß and G. Smolka. “Attributive concept descriptions with complements.” In: *Artificial Intelligence* 48.1 (1991), pp. 1–26. ISSN: 0004-3702. DOI: https://doi.org/10.1016/0004-3702(91)90078-X. URL: https://www.sciencedirect.com/science/article/pii/000437029190078X.

[340] L. Schneider. “How to Build a Foundational Ontology.” In: *KI 2003: Advances in Artificial Intelligence*. Ed. by A. Günter, R. Kruse, and B. Neumann. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 120–134. ISBN: 978-3-540-39451-8.

[341] T. Schneider and M. Šimkus. “Ontologies and Data Management: A Brief Survey.” In: *KI - Künstliche Intelligenz* 34 (Aug. 2020). DOI: 10.1007/s13218-020-00686-3.

[342] M. Schorlemmer and Y. Kalfoglou. "Institutionalising Ontology-Based Semantic Integration." In: *Appl. Ontol.* 3.3 (July 2008), pp. 131–150. ISSN: 1570-5838.

[343] S. Seppälä, B. Smith, and W. Ceusters. "Applying the Realism-Based Ontology-Versioning Method for Tracking Changes in the Basic Formal Ontology." In: *Formal Ontology in Information Systems (FOIS 2014)*. Ed. by P. Garbacz and O. Kutz. IOS Press, 2014, pp. 227–240.

[344] A. Sernadas, C. Sernadas, and C. Caleiro. "Fibring of logics as a categorial construction." In: *Journal of Logic and Computation* 9.2 (Apr. 1999), pp. 149–179. ISSN: 0955-792X. DOI: 10.1093/logcom/9.2.149. eprint: https://academic.oup.com/logcom/article-pdf/9/2/149/2892952/090149.pdf. URL: https://doi.org/10.1093/logcom/9.2.149.

[345] A. Sernadas, C. Sernadas, and J. Rasga. "On meet-combination of logics." In: *Journal of Logic and Computation* 22.6 (2012), pp. 1453–1470. DOI: 10.1093/logcom/exr035.

[346] A. Sernadas et al. "on Graph-Theoretic Fibring of Logics." In: *Journal of Logic and Computation* 19.6 (2009), pp. 1321–1357.

[347] A. Sernadas, C. Sernadas, and A. Zanardo. "Fibring Modal First-Order Logics: Completeness Preservation." In: *Logic Journal of the IGPL* 10.4 (July 2002), pp. 413–451. ISSN: 1367-0751. DOI: 10.1093/jigpal/10.4.413. eprint: https://academic.oup.com/jigpal/article-pdf/10/4/413/2321115/100413.pdf. URL: https://doi.org/10.1093/jigpal/10.4.413.

[348] A. Sernadas et al. "A graph-theoretic account of logics." In: *Journal of Logic and Computation* 19.6 (2009), pp. 1281–1320.

[349] C. Sernadas, J. Rasga, and W. A. Carnielli. "Modulated Fibring and the Collapsing Problem." In: *The Journal of Symbolic Logic* 67.4 (2002), pp. 1541–1569. ISSN: 00224812. URL: http://www.jstor.org/stable/3648586 (visited on 01/08/2024).

[350] S. Shapiro. *Vagueness in Context*. New York: Oxford University Press, 2006.

[351] S. Shapiro. *Varieties of Logic*. Oxford and New York: Oxford University Press, 2014.

[352] A. Sheth. *Forging Trust in Tomorrow's AI: A Roadmap for Reliable, Explainable, and Safe NeuroSymbolic Systems*. https://ontologforum.com/index.php/ConferenceCall_2024_04_17. Accessed: 2024–08-11. 2024.

[353] A. Sheth, K. Roy, and M. Gaur. "Neurosymbolic Artificial Intelligence (Why, What, and How)." In: *IEEE Intelligent Systems* 38.3 (2023), pp. 56–62. DOI: 10.1109/MIS.2023.3268724.

[354] P. Simons. *Parts: A Study in Ontology*. New York: Oxford University Press, 1987.

[355] E. Sirin and B. Parsia. "Pellet: An owl dl reasoner." In: *Proc. of the 2004 Description Logic Workshop (DL 2004)*. 2004, pp. 212–213.

[356] J. Smid. "The logic behind Quine's criterion of ontological commitment." In: *European Journal of Philosophy* 28.3 (2020), pp. 789–804. DOI: https://doi.org/10.1111/ejop.12534. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/ejop.12534. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/ejop.12534.

[357] B. Smith. “An Essay in Formal Ontology.” In: *Grazer Philosophische Studien* 6.1 (1978), pp. 39–62. DOI: 10.5840/gps197864.

[358] B. Smith. “Ontology.” In: *The Blackwell Guide to the Philosophy of Computing and Information*. John Wiley & Sons, Ltd, 2004. Chap. 11, pp. 153–166. ISBN: 9780470757017. DOI: https://doi.org/10.1002/9780470757017.ch11. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470757017.ch11. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470757017.ch11.

[359] B. Smith and K. Mulligan. “Framework for Formal Ontology.” In: *Topoi* 2.1 (1983), pp. 73–85. DOI: 10.1007/bf00139703.

[360] B. Smith and J. Sims. “Revisiting the Derrida Affair.” In: *Sophia* 38.2 (1999), pp. 142–169. DOI: 10.1007/bf02786336.

[361] B. Smith and C. Welty. “Ontology: Towards a New Synthesis.” In: Jan. 2001, pp. iii–x.

[362] B. Smith et al. “The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration.” In: *Nature biotechnology* 25.11 (2007), pp. 1251–1255.

[363] M. S. G. Smolka. “Attributive Concept Descriptions with Complements.” In: *Artificial Intelligence* 48.1 (1991), pp. 1–26. DOI: 10.1016/0004-3702(91)90078-x.

[364] R. Smullyan. *First-order Logic*. Dover books on advanced mathematics. Dover, 1995. ISBN: 9780486683706. URL: https://books.google.com.br/books?id=kgvhQ-oSZiUC.

[365] J. Sowa. *Without Ontology, LLMs are clueless*. https://ontologforum.com/index.php/ConferenceCall_2024_02_28. Accessed: 2024–08-11. 2024.

[366] J. F. Sowa. “Conceptual graphs as a universal knowledge representation.” In: *Computers & Mathematics with Applications* 23.2-5 (1992), pp. 75–93.

[367] J. F. Sowa. “Top-level ontological categories.” In: *International Journal of Human-Computer Studies* 43.5 (1995), pp. 669–685. ISSN: 1071-5819. DOI: https://doi.org/10.1006/ijhc.1995.1068. URL: https://www.sciencedirect.com/science/article/pii/S1071581985710683.

[368] M. Spies and C. Roche. “Aristotelian Ontologies and Owl Modeling.” In: *Wspi 2006: Contributions to the Third International Workshop on Philosophy and Informatics*. Ed. by I. Johansson, B. Klein, and T. Roth-Berghofer. 2006.

[369] D. I. Spivak. *Category theory for the sciences*. MIT press, 2014.

[370] P. Spyns, Y. Tang, and R. Meersman. “An ontology engineering methodology for DOGMA.” In: *Applied Ontology* 3.1-2 (2008), pp. 13–39.

[371] A. Stadnicki, F. Filip Pietroń, and P. Burek. “Towards a Modern Ontology Development Environment.” In: *Procedia Computer Science* 176 (2020). Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 24th International Conference KES2020, pp. 753–762. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2020.09.070. URL: https://www.sciencedirect.com/science/article/pii/S1877050920319657.

[372] G. Stoilos et al. "A novel approach and practical algorithms for ontology integration." In: *The Semantic Web–ISWC 2018: 17th International Semantic Web Conference, Monterey, CA, USA, October 8–12, 2018, Proceedings, Part I 17*. Springer. 2018, pp. 458–476.

[373] G. Stoilos et al. "Reasoning with very expressive fuzzy description logics." In: *Journal of Artificial Intelligence Research* 30 (2007), pp. 273–320.

[374] M. Stonebraker. "Chapter 12: A Biased Take on a Moving Target: Data Integration." In: *Readings in Database Systems, 5th Edition*. Ed. by P. Bailis, J. M. Hellerstein, and M. Stonebraker. MIT Press, 2015.

[375] H. Stuckenschmidt, C. Parent, and S. Spaccapietra. *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*. 1st. Springer Publishing Company, Incorporated, 2009. ISBN: 3642019064.

[376] R. Studer, V. Benjamins, and D. Fensel. "Knowledge engineering: Principles and methods." In: *Data & Knowledge Engineering* 25.1 (1998), pp. 161–197. ISSN: 0169-023X. DOI: https://doi.org/10.1016/S0169-023X(97)00056-6. URL: https://www.sciencedirect.com/science/article/pii/S0169023X97000566.

[377] P. Studtmann. "Aristotle's Categories." In: *The Stanford Encyclopedia of Philosophy*. Ed. by E. N. Zalta and U. Nodelman. Spring 2024. Metaphysics Research Lab, Stanford University, 2024.

[378] D. Tajer and C. Fiore. "Logical Pluralism and Interpretations of Logical Systems." In: *Logic and Logical Philosophy* 31 (2022), pp. 1–26. DOI: 10.12775/llp.2022.007.

[379] L. Tarán. *Parmenides: A Text with Translation, Commentary, and Critical Essays*. Princeton University Press, 1965. ISBN: 9780691071282. URL: https://books.google.com.br/books?id=Ho5zvwEACAAJ.

[380] A. Tarski. "Fundamentale begriffe der methodologie der deduktiven wissenschaften. i." In: *Monatshefte für Mathematik und Physik* 37 (1930), pp. 361–404.

[381] A. Tarski et al. "On the concept of logical consequence." In: *Logic, semantics, metamathematics* 52 (1936), pp. 409–420.

[382] B. A. Trakhtenbrot. "Impossibility of an algorithm for the decision problem in finite classes." In: *Doklady Akademii Nauk SSSR* 70.4 (1950), pp. 569–572.

[383] D. Tsarkov and I. Horrocks. "FaCT++ Description Logic Reasoner: System Description." In: *Automated Reasoning*. Ed. by U. Furbach and N. Shankar. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 292–297. ISBN: 978-3-540-37188-5.

[384] D. Tsarkov and I. Horrocks. "FaCT++ description logic reasoner: System description." In: *International joint conference on automated reasoning*. Springer. 2006, pp. 292–297.

[385] A. Turing. "On Computable Numbers, with an Application to the Entscheidungsproblem." In: *Proceedings of the London Mathematical Society* 42.1 (1936), pp. 230–265. DOI: 10.2307/2268810.

[386] K. Twardowski. "Introduction." In: *On the Content and Object of Presentations: A Psychological Investigation*. Ed. by H. Bergmann. M. Nijhoff, 1977.

[387] M. Y. Vardi. *Why is modal logic so robustly decidable?* Tech. rep. 1997.

[388] A. C. Varzi. "47Counting and Countenancing." In: *Composition as Identity*. Oxford University Press, Aug. 2014. ISBN: 9780199669615. DOI: 10.1093/acprof:oso/9780199669615.003.0003. eprint: https://academic.oup.com/book/0/chapter/150268158/chapter-ag-pdf/44982464/book_6442_section_150268158.ag.pdf. URL: https://doi.org/10.1093/acprof:oso/9780199669615.003.0003.

[389] A. C. Varzi. "On Logical Relativity." In: *Philosophical Issues* 12.1 (2002), pp. 197–219. DOI: 10.1111/j.1758-2237.2002.tb00067.x.

[390] A. Vaswani et al. "Attention is All you Need." In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[391] V. Viljanen. "Spinoza's Ontology." In: *The Cambridge Companion to Spinoza's Ethics*. Ed. by O. Koistinen. Cambridge Companions to Philosophy. Cambridge University Press, 2009, pp. 56–78. DOI: 10.1017/CCOL9780521853392.004.

[392] P. R. Visser et al. "An analysis of ontological mismatches: Heterogeneity versus interoperability." In: *AAAI 1997 Spring Symposium on Ontological Engineering*. Vol. 89. Stanford, USA. 1997.

[393] J. Von Neumann. "The mathematician." In: *The works of the mind* 1.1 (1947), pp. 180–196.

[394] M. Warglien and A. C. Varzi. "The Geometry of Negation." In: *Journal of Applied Non-Classical Logics* 13.1 (2003), pp. 9–19. DOI: 10.3166/jancl.13.9-19.

[395] T. Wei et al. "The TAO CI Ontology of Vases of the Ming and Qing Dynasties." In: *Appl. Ontol.* 17.3 (Jan. 2022), pp. 423–441. ISSN: 1570-5838. DOI: 10.3233/AO-220270. URL: https://doi.org/10.3233/AO-220270.

[396] C. Welty and W. Andersen. "Towards OntoClean 2.0: A framework for rigidity." In: *Applied Ontology* 1 (Jan. 2005), pp. 107–116.

[397] W. Wong, W. Liu, and M. Bennamoun. "Ontology learning from text: A look back and into the future." In: *ACM Comput. Surv.* 44.4 (Sept. 2012). ISSN: 0360-0300. DOI: 10.1145/2333112.2333115. URL: https://doi.org/10.1145/2333112.2333115.

[398] C. Woods et al. "An Ontology for Maintenance Procedure Documentation." In: *Applied ontology* (forthcoming), pp. 1–38. DOI: 10.3233/ao-230279.

[399] R. Woodward. "119The Questions of Ontology." In: *The Language of Ontology*. Oxford University Press, June 2021. ISBN: 9780192895332. DOI: 10.1093/oso/9780192895332.003.0008. eprint: https://academic.oup.com/book/0/chapter/340312379/chapter-pdf/42886456/oso-9780192895332-chapter-8.pdf. URL: https://doi.org/10.1093/oso/9780192895332.003.0008.

[400] X. Wu et al. *Differentiable Fuzzy $\mathcal{ALC}$: A Neural-Symbolic Representation Language for Symbol Grounding*. 2022. arXiv: 2211.12006 [cs.AI].

[401] A. Zanardo, A. Sernadas, and C. Sernadas. "Fibring: Completeness Preservation." In: *The Journal of Symbolic Logic* 66.1 (2001), pp. 414–439. ISSN: 00224812. URL: http://www.jstor.org/stable/2694931 (visited on 02/06/2024).

[402] A. V. Zhdanova and U. Keller. "Choosing an ontology language." In: *International Journal of Computer and Information Engineering* 1.4 (2007), pp. 1011–1014.

[403] A. Zimmermann et al. "Formalizing Ontology Alignment and its Operations with Category Theory." In: vol. 150. Nov. 2006, pp. 277–288.

Appendix A

Example Course Ontology

This appendix presents the RDF representation of the example "course **o**ntology" discussed in section 7.4. The **o**ntology was created through Protégé [270].

```
<?xml version="1.0"?>
<rdf:RDF xmlns="http://www.semanticweb.org/gabriel/ontologies/2024/6/
    example-course-ontology#"
         xml:base="http://www.semanticweb.org/gabriel/ontologies
    /2024/6/example-course-ontology"
         xmlns:owl="http://www.w3.org/2002/07/owl#"
         xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:xml="http://www.w3.org/XML/1998/namespace"
         xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
         xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
         xmlns:example-course-ontology="http://www.semanticweb.org/
    gabriel/ontologies/2024/6/example-course-ontology#">
     <owl:Ontology rdf:about="http://www.semanticweb.org/gabriel/
    ontologies/2024/6/example-course-ontology"/>



     <!--
     ///////////
     //
     // Object Properties
     //
     ///////////
         -->




     <!-- http://www.semanticweb.org/gabriel/ontologies/2024/6/example
    -course-ontology#takes -->

     <owl:ObjectProperty rdf:about="http://www.semanticweb.org/gabriel
    /ontologies/2024/6/example-course-ontology#takes">
```

```xml
        <rdfs:subPropertyOf rdf:resource="http://www.w3.org/2002/07/owl#topObjectProperty"/>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#AsymmetricProperty"/>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#IrreflexiveProperty"/>
    </owl:ObjectProperty>



    <!-- http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#teaches -->

    <owl:ObjectProperty rdf:about="http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#teaches">
        <rdfs:subPropertyOf rdf:resource="http://www.w3.org/2002/07/owl#topObjectProperty"/>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#AsymmetricProperty"/>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#IrreflexiveProperty"/>
    </owl:ObjectProperty>



    <!--
    ///////////
    //
    // Classes
    //
    ///////////
        -->




    <!-- http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#Course -->

    <owl:Class rdf:about="http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#Course"/>



    <!-- http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#Person -->

    <owl:Class rdf:about="http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#Person"/>


```

```

    <!-- http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#Professor -->

    <owl:Class rdf:about="http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#Professor">
        <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#Person"/>
    </owl:Class>



    <!-- http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#Student -->

    <owl:Class rdf:about="http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#Student">
        <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#Person"/>
    </owl:Class>



    <!--
    ///////////
    //
    // Individuals
    //
    ///////////
        -->




    <!-- http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#CourseA -->

    <owl:NamedIndividual rdf:about="http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#CourseA">
        <rdf:type rdf:resource="http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#Course"/>
    </owl:NamedIndividual>



    <!-- http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#ProfessorA -->

    <owl:NamedIndividual rdf:about="http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#ProfessorA">
```

```
        <rdf:type rdf:resource="http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#Professor"/>
        <owl:differentFrom rdf:resource="http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#ProfessorB"/>
        <teaches rdf:resource="http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#CourseA"/>
    </owl:NamedIndividual>



    <!-- http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#ProfessorB -->

    <owl:NamedIndividual rdf:about="http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#ProfessorB">
        <rdf:type rdf:resource="http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#Professor"/>
        <rdf:type rdf:resource="http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#Student"/>
        <owl:differentFrom rdf:resource="http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#StudentA"/>
        <takes rdf:resource="http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#CourseA"/>
    </owl:NamedIndividual>



    <!-- http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#StudentA -->

    <owl:NamedIndividual rdf:about="http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#StudentA">
        <rdf:type rdf:resource="http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#Student"/>
        <takes rdf:resource="http://www.semanticweb.org/gabriel/ontologies/2024/6/example-course-ontology#CourseA"/>
    </owl:NamedIndividual>
</rdf:RDF>



<!-- Generated by the OWL API (version 4.5.29.2024-05-13T12:11:03Z) https://github.com/owlcs/owlapi -->
```

Listing A.1: Example course **o**ntology in RDF.

Appendix B

Basic Formal Ontology

This is an intermediate representation of the Basic Formal Ontology (BFO)[1]. The **o**ntology representation was generated through Protégé [270].

Classes

ConnectedSpatiotemporalRegion

ConnectedSpatiotemporalRegion ≡ SpatiotemporalInstant ⊔ SpatiotemporalInterval

ConnectedSpatiotemporalRegion ⊑ SpatiotemporalRegion

ConnectedSpatiotemporalRegion ⊑ ¬ ScatteredSpatiotemporalRegion

ConnectedTemporalRegion

ConnectedTemporalRegion ≡ TemporalInstant ⊔ TemporalInterval

ConnectedTemporalRegion ⊑ TemporalRegion

ConnectedTemporalRegion ⊑ ¬ ScatteredTemporalRegion

Continuant

Continuant ≡ DependentContinuant ⊔ IndependentContinuant ⊔ SpatialRegion

Continuant ⊑ Entity

Continuant ⊑ ¬ Occurrent

DependentContinuant

DependentContinuant ≡ GenericallyDependentContinuant ⊔ SpecificallyDependentContinuant

DependentContinuant ⊑ Continuant

[1]The **o**ntology was originally retrieved from and is accessible at http://www.ifomis.org/bfo/1.1.

$DependentContinuant \sqsubseteq \neg\ IndependentContinuant$

$DependentContinuant \sqsubseteq \neg\ SpatialRegion$

Disposition

$Disposition \sqsubseteq RealizableEntity$

$Disposition \sqsubseteq \neg\ Role$

$Disposition \sqsubseteq \neg\ Function$

Entity

$Entity \equiv Continuant \sqcup Occurrent$

FiatObjectPart

$FiatObjectPart \sqsubseteq IndependentContinuant$

$FiatObjectPart \sqsubseteq \neg\ ObjectBoundary$

$FiatObjectPart \sqsubseteq \neg\ Object$

$FiatObjectPart \sqsubseteq \neg\ ObjectAggregate$

$FiatObjectPart \sqsubseteq \neg\ Site$

FiatProcessPart

$FiatProcessPart \sqsubseteq ProcessualEntity$

$FiatProcessPart \sqsubseteq \neg\ ProcessAggregate$

$FiatProcessPart \sqsubseteq \neg\ ProcessualContext$

$FiatProcessPart \sqsubseteq \neg\ ProcessBoundary$

$FiatProcessPart \sqsubseteq \neg\ Process$

Function

$Function \sqsubseteq RealizableEntity$

$Function \sqsubseteq \neg\ Role$

$Function \sqsubseteq \neg\ Disposition$

GenericallyDependentContinuant

$GenericallyDependentContinuant \sqsubseteq DependentContinuant$

$GenericallyDependentContinuant \sqsubseteq \neg\ SpecificallyDependentContinuant$

IndependentContinuant

IndependentContinuant $\equiv$ FiatObjectPart $\sqcup$ Object $\sqcup$ ObjectAggregate $\sqcup$ ObjectBoundary $\sqcup$ Site

IndependentContinuant $\sqsubseteq$ Continuant

IndependentContinuant $\sqsubseteq$ $\neg$ DependentContinuant

IndependentContinuant $\sqsubseteq$ $\neg$ SpatialRegion

Object

Object $\sqsubseteq$ IndependentContinuant

Object $\sqsubseteq$ $\neg$ ObjectBoundary

Object $\sqsubseteq$ $\neg$ FiatObjectPart

Object $\sqsubseteq$ $\neg$ Site

Object $\sqsubseteq$ $\neg$ ObjectAggregate

ObjectAggregate

ObjectAggregate $\sqsubseteq$ IndependentContinuant

ObjectAggregate $\sqsubseteq$ $\neg$ ObjectBoundary

ObjectAggregate $\sqsubseteq$ $\neg$ Site

ObjectAggregate $\sqsubseteq$ $\neg$ FiatObjectPart

ObjectAggregate $\sqsubseteq$ $\neg$ Object

ObjectBoundary

ObjectBoundary $\sqsubseteq$ IndependentContinuant

ObjectBoundary $\sqsubseteq$ $\neg$ Site

ObjectBoundary $\sqsubseteq$ $\neg$ ObjectAggregate

ObjectBoundary $\sqsubseteq$ $\neg$ Object

ObjectBoundary $\sqsubseteq$ $\neg$ FiatObjectPart

Occurrent

Occurrent $\equiv$ ProcessualEntity $\sqcup$ SpatiotemporalRegion $\sqcup$ TemporalRegion

Occurrent $\sqsubseteq$ Entity

Occurrent $\sqsubseteq$ $\neg$ Continuant

OneDimensionalRegion

OneDimensionalRegion $\sqsubseteq$ SpatialRegion

OneDimensionalRegion $\sqsubseteq \neg$ ZeroDimensionalRegion

OneDimensionalRegion $\sqsubseteq \neg$ TwoDimensionalRegion

OneDimensionalRegion $\sqsubseteq \neg$ ThreeDimensionalRegion

Process

Process $\sqsubseteq$ ProcessualEntity

Process $\sqsubseteq \neg$ ProcessBoundary

Process $\sqsubseteq \neg$ ProcessAggregate

Process $\sqsubseteq \neg$ ProcessualContext

Process $\sqsubseteq \neg$ FiatProcessPart

ProcessAggregate

ProcessAggregate $\sqsubseteq$ ProcessualEntity

ProcessAggregate $\sqsubseteq \neg$ FiatProcessPart

ProcessAggregate $\sqsubseteq \neg$ ProcessBoundary

ProcessAggregate $\sqsubseteq \neg$ Process

ProcessAggregate $\sqsubseteq \neg$ ProcessualContext

ProcessBoundary

ProcessBoundary $\sqsubseteq$ ProcessualEntity

ProcessBoundary $\sqsubseteq \neg$ FiatProcessPart

ProcessBoundary $\sqsubseteq \neg$ ProcessAggregate

ProcessBoundary $\sqsubseteq \neg$ ProcessualContext

ProcessBoundary $\sqsubseteq \neg$ Process

ProcessualContext

ProcessualContext $\sqsubseteq$ ProcessualEntity

ProcessualContext $\sqsubseteq \neg$ FiatProcessPart

ProcessualContext $\sqsubseteq \neg$ ProcessBoundary

ProcessualContext $\sqsubseteq \neg$ Process

ProcessualContext $\sqsubseteq \neg$ ProcessAggregate

ProcessualEntity

ProcessualEntity $\equiv$ FiatProcessPart $\sqcup$ Process $\sqcup$ ProcessAggregate $\sqcup$ ProcessBoundary $\sqcup$ ProcessualContext

ProcessualEntity $\sqsubseteq$ Occurrent

ProcessualEntity $\sqsubseteq \neg$ TemporalRegion

ProcessualEntity $\sqsubseteq \neg$ SpatiotemporalRegion

Quality

Quality $\sqsubseteq$ SpecificallyDependentContinuant

Quality $\sqsubseteq \neg$ RealizableEntity

RealizableEntity

RealizableEntity $\sqsubseteq$ SpecificallyDependentContinuant

RealizableEntity $\sqsubseteq \neg$ Quality

Role

Role $\sqsubseteq$ RealizableEntity

Role $\sqsubseteq \neg$ Disposition

Role $\sqsubseteq \neg$ Function

ScatteredSpatiotemporalRegion

ScatteredSpatiotemporalRegion $\sqsubseteq$ SpatiotemporalRegion

ScatteredSpatiotemporalRegion $\sqsubseteq \neg$ ConnectedSpatiotemporalRegion

ScatteredTemporalRegion

ScatteredTemporalRegion $\sqsubseteq$ TemporalRegion

ScatteredTemporalRegion $\sqsubseteq \neg$ ConnectedTemporalRegion

Site

Site $\sqsubseteq$ IndependentContinuant

Site $\sqsubseteq \neg$ ObjectBoundary

Site $\sqsubseteq \neg$ ObjectAggregate

Site $\sqsubseteq \neg$ Object

Site $\sqsubseteq \neg$ FiatObjectPart

SpatialRegion

SpatialRegion $\equiv$ OneDimensionalRegion $\sqcup$ ThreeDimensionalRegion $\sqcup$ TwoDimensionalRegion $\sqcup$ ZeroDimensionalRegion

SpatialRegion $\sqsubseteq$ Continuant

SpatialRegion $\sqsubseteq \neg$ DependentContinuant

SpatialRegion $\sqsubseteq \neg$ IndependentContinuant

SpatiotemporalInstant

SpatiotemporalInstant $\sqsubseteq$ ConnectedSpatiotemporalRegion

SpatiotemporalInstant $\sqsubseteq \neg$ SpatiotemporalInterval

SpatiotemporalInterval

SpatiotemporalInterval $\sqsubseteq$ ConnectedSpatiotemporalRegion

SpatiotemporalInterval $\sqsubseteq \neg$ SpatiotemporalInstant

SpatiotemporalRegion

SpatiotemporalRegion $\equiv$ ConnectedSpatiotemporalRegion $\sqcup$ ScatteredSpatiotemporalRegion

SpatiotemporalRegion $\sqsubseteq$ Occurrent

SpatiotemporalRegion $\sqsubseteq \neg$ TemporalRegion

SpatiotemporalRegion $\sqsubseteq \neg$ ProcessualEntity

SpecificallyDependentContinuant

SpecificallyDependentContinuant $\equiv$ Quality $\sqcup$ RealizableEntity

SpecificallyDependentContinuant $\sqsubseteq$ DependentContinuant

SpecificallyDependentContinuant $\sqsubseteq \neg$ GenericallyDependentContinuant

TemporalInstant

TemporalInstant $\sqsubseteq$ ConnectedTemporalRegion

TemporalInstant $\sqsubseteq \neg$ TemporalInterval

TemporalInterval

TemporalInterval $\sqsubseteq$ ConnectedTemporalRegion

TemporalInterval $\sqsubseteq \neg$ TemporalInstant

TemporalRegion

TemporalRegion ≡ ConnectedTemporalRegion ⊔ ScatteredTemporalRegion

TemporalRegion ⊑ Occurrent

TemporalRegion ⊑ ¬ SpatiotemporalRegion

TemporalRegion ⊑ ¬ ProcessualEntity

ThreeDimensionalRegion

ThreeDimensionalRegion ⊑ SpatialRegion

ThreeDimensionalRegion ⊑ ¬ OneDimensionalRegion

ThreeDimensionalRegion ⊑ ¬ ZeroDimensionalRegion

ThreeDimensionalRegion ⊑ ¬ TwoDimensionalRegion

TwoDimensionalRegion

TwoDimensionalRegion ⊑ SpatialRegion

TwoDimensionalRegion ⊑ ¬ OneDimensionalRegion

TwoDimensionalRegion ⊑ ¬ ZeroDimensionalRegion

TwoDimensionalRegion ⊑ ¬ ThreeDimensionalRegion

ZeroDimensionalRegion

ZeroDimensionalRegion ⊑ SpatialRegion

ZeroDimensionalRegion ⊑ ¬ OneDimensionalRegion

ZeroDimensionalRegion ⊑ ¬ ThreeDimensionalRegion

ZeroDimensionalRegion ⊑ ¬ TwoDimensionalRegion

Object properties

Data properties

Individuals

Datatypes

PlainLiteral