THE UNIVERSITY *of* EDINBURGH

Ontology Evolution in Physics

Michael Chan

Doctor of Philosophy

Centre for Intelligent Systems and their Applications
School of Informatics
University of Edinburgh

2013

Abstract

With the advent of reasoning problems in dynamic environments, there is an increasing need for automated reasoning systems to automatically adapt to unexpected changes in representations. In particular, the automation of the evolution of their ontologies needs to be enhanced without substantially sacrificing expressivity in the underlying representation. Revision of beliefs is not enough, as adding to or removing from beliefs does not change the underlying formal language. General reasoning systems employed in such environments should also address situations in which the language for representing knowledge is not shared among the involved entities, e.g., the ontologies in a multi-ontology environment or the agents in a multi-agent environment. Our techniques involve diagnosis of faults in existing, possibly heterogeneous, ontologies and then resolution of these faults by manipulating the signature and/or the axioms.

This thesis describes the design, development and evaluation of GALILEO (*Guided Analysis of Logical Inconsistencies Lead to Evolution of Ontologies*), a system designed to detect conflicts in highly expressive ontologies and resolve the detected conflicts by performing appropriate repair operations. The integrated mechanism that handles ontology evolution is able to distinguish between various types of conflicts, each corresponding to a unique kind of ontological fault. We apply and develop our techniques in the domain of Physics. This an excellent domain because many of its seminal advances can be seen as examples of ontology evolution, i.e. changing the way that physicists perceive the world, and case studies are well documented – unlike many other domains. Our research covers analysing a wide ranging development set of case studies and evaluating the performance of the system on a test set. Because the formal representations of most of the case studies are non-trivial and the underlying logic has a high degree of expressivity, we face some tricky technical challenges, including dealing with the potentially large number of choices in diagnosis and repair. In order to enhance the practicality and the manageability of the ontology evolution process, GALILEO incorporates the functionality of generating physically meaningful diagnoses and repairs and, as a result, narrowing the search space to a manageable size.

Acknowledgements

Choosing the University of Edinburgh for my Ph.D. program was one of the best decisions I have ever made. For the last few years, I have enjoyed the exceptionally high concentration of friendly, helpful and talented people.

Not surprisingly, this thesis could not have been produced without the support of many people. This research was funded by EPSRC grant EP/G000700/1 and by an Overseas Research Studentship (ORS) Award, and I thank my supervisors: Alan Bundy and Jos Lehmann. I am extremely grateful and glad to have a supervisor who is as erudite, imaginative, dedicated, energetic and astute as Alan. He should be acknowledged as the best supervisor one can have. Without his critical and insightful comments and his careful reading and rereading of drafts, the thesis could not be in the shape it is in. I am also very grateful to Jos for his tireless attention to the details involved in the research. I would also like to thank all the members of the DReaM group for providing much support, help and encouragement.

My thanks extend to Alexander Krauss and Tobias Nipkow for sharing their knowledge and their interests in my project. Thank you to Alan Smaill for proof reading parts of this thesis and for his detailed and constructive feedback throughout the project.

Finally, my deepest gratitude goes to my family for their unflagging love and support throughout my life. Their support has been unconditional and constant all these years.

Declaration

I declare that this thesis was composed by myself, that a significant majority of the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified. Individual elements of the work that were substantially contributed by myself include the implementation and the design of the system and the concrete analysis and experiments that answer the research hypotheses.

(*Michael Chan*)

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

AI and, more generally, computer science are presently faced with the challenge that intelligent agents must be able to represent and manipulate their own knowledge. For an agent to perform reasoning, the conceptualisation of the entities in a domain of discourse is usually represented in the agent's *knowledge-base*. Unfortunately, as outlined by Bundy and McNeill (2006), the world is inherently dynamic, so changes in our conceptualisation of this infinitely-complex domain are necessary. Knowledge-base evolution, the process of updating a knowledge-base when new information is acquired, poses significant challenges to the representation of knowledge and the formalisation of reasoning by, for example, introducing inconsistency, ambiguity, and incompleteness into the knowledge-base. There are several specific types of knowledge-base evolution, including database schema evolution and, the focus of this thesis, *ontology evolution*. The term *ontology* originated in Philosophy, where it is the philosophical study of being, which covers the nature of existence and the structure of reality. The word "ontology" has been adopted and adapted within Computing as meaning a formal representation of the concepts within a domain and the relationships between those concepts.

Defined by Stojanovic et al. (2002), ontology evolution is "the timely adaptation of an ontology to the arisen changes and the consistent propagation of these changes to dependent artefacts". Although Stojanovic et al.'s focus is on user-driven approaches, we believe that the definition is also appropriate for *automated* ontology evolution. Interactive manipulation has a number of limitations: for instance, the process of identifying parts of the ontology appropriate for manipulation can be difficult and arduous even to expert users, and if a manipulation is required during agent communication, all further actions must be blocked until manual intervention takes place. Since scaling interactive manipulation can be tremendously challenging, mechanisms for automated ontology evolution are highly desirable.

The need to reliably handle ontology evolution has been intensified by the increased demands created by multi-agent systems, for example, the vision of a vast number of agents interacting in the Semantic Web. Investigations in ontology evolution will elucidate new mechanisms for more accurate agent communication and give a better understanding of the general knowledge-base evolution process. If autonomous systems are to deal with an ever-changing world, they must be able to autonomously update their own ontologies. This requires them to be able to both detect faults in and manipulate their theory of the world. The failure to update the theory accordingly will usually lead to ineffective or inconsistent communication, caused by the discrepancy between different views or conceptualisations of the world. Such updates must go beyond the ability to change beliefs and learn new concepts in terms of the old ones, and in these cases the underlying syntax and semantics of the ontologies themselves may also require alterations. For instance, if sales tax is to be introduced into an inventory system, then the inventory agent may want to revise the original signature of the function returning the price of an item in order to account for sales tax. One appropriate solution would be to increase the arity by providing an additional argument for indicating the corresponding amount of sales tax. This new argument implies that the total price of an item varies with sales tax, which meets the new requirement.

This project investigates the techniques that may be needed to be built into a system that facilitates automated ontology evolution. We experiment with these techniques in a system which we call GALILEO (*Guided Analyses of Logical Inconsistencies Lead to Evolution of Ontologies*). To identify and develop such techniques, we depart away from working in everyday, common domains and investigate records of ontology evolution in the history of Physics (Bundy and Chan, 2008). Many of the most seminal advances in the development of Physics required some form of ontology evolution.

1.1 Motivation

Epistemology, the study of knowledge, has a long history in philosophy. Initiated by Descartes and Locke, the work during the early era was based on intuitive, rationalistic and introspective modes of reasoning. To eliminate subjectivism, Leibniz proposed a need for a rigorous formalisation of reasoning that would allow errors to be as easily detected as in arithmetic. Leibniz's idea of enhancing the role of mathematics and using a logical structure of reasoning evolved to become the concept of a universal scientific language, known as *Characteristica Universalis*. Other concepts, such as Kant's system of categories, have also demonstrated the power of formal representation and logical analysis of reasoning.

The perspective of a formal approach has been widely accepted and supported by the contemporary scientific community at large. Such a rigorous approach can provide analysis of reasoning with precision and reliability; consequently, it can benefit all areas concerned with the role and state of knowledge. Motivated by various applications in the past couple of decades, significantly more attention has been drawn from researchers in various fields ranging from philosophy, economics, and linguistics to theoretical computer science and artificial intelligence. The research foci across these fields are generally different, but it is typically conceded that appropriate representation of knowledge is crucial for effective reasoning, as argued by Pólya (1945).

A formal approach to handling ontology evolution automatically is now urgent, due to the demand created by multi-agent systems. Ontological conflicts between agents are central to breakdowns in agent communication, as even the mundane assumptions each agent makes about one another, e.g., common symbols refer to the same concept, may no longer hold when such conflicts arise. These conflicts are not only caused by a disagreement about the interpretation of the world, which is the central focus of *belief revision*, but by the disagreement on the representation of the knowledge. In complex and dynamic agent environments, it is not reasonable to even assume that a shared concept must have the same representation, because a discrepant representation may be the result of making different modelling assumptions. This problem is beyond the scope of techniques for *ontology matching* and *mapping*, because reusing existing concepts in the ontology is often insufficient due to the inadequacy of the original representation. Thus, in order to facilitate more robust agent communication, the techniques for ontology evolution adopted must be able to automatically repair the *signature* of the ontologies, which includes changing the existing representation and inventing new concepts to help give the repaired ontology more (accurate) meaning. This process may be seen as being analogous to biological evolution, because the ontology automatically adapts to its changing environment. The system developed, GALILEO, can diagnose specific conflicts between ontologies and automatically repair the ontologies so that the detected conflict is resolved.

1.2 Methodology

As described in §2.4, there are many types of ontology change: ontology evolution, mapping, morphism, and matching. Ontology evolution is considered to be the most radical form as it manipulates the profound structure of the ontology (Flouris et al., 2008). As opposed to the other types, ontology evolution implements changes to both the syntax and the semantics of the ontology. Automated ontology evolution explores how the theory should evolve given some new conflicting information and how an agent adapts to the changing world and goals. Typically, the original theory itself is consistent (locally consistent), but the merge with the newly obtained information leads to inconsistency (globally inconsistent) or a representational conflict. For instance, if the observed value of a function unexpectedly varies when the theory predicts it to remain constant, then the theory should be mended to consider the parameter causing the variation as an argument of the function. There are many other similar reasoning faults that may occur in problem solving, and they are eventually resolved by the solver's intelligence and creativity.

In this project, we have explored a) patterns for detecting ontological faults arising from taking into account some new information, and b) transformation rules for resolving the resulting conflicts between concepts – we call the composition of such patterns and transformation rules *ontology repair plans* (ORPs). Ontology repair plans are generic combinations of diagnosis and repair operations that guide the ontology evolution process. To identify such patterns, we do not follow the relatively usual research direction to study everyday, common domains, e.g., operating a shop, managing a library, and so forth, which we believe are excessively open and lack of structure. Since our goal is to obtain an understanding of the fundamental requirements for the realisation of automated ontology evolution, the domain of interest must be chosen with care such that examples of ontology evolution are reasonably accessible and that a kind of formalism already underlies the definitions of relevant concepts. Thus, we investigate records of ontology evolution in the history of Physics; we believe capturing human reasoners' imagination used in Physics problem solving can help develop and advance mechanisms for ontology evolution. Physics concepts are usually mathematically defined and the evolutionary process in Physics is generally well documented, as it attracts significant interest from both the scientific community, the press and historians. Detailed accounts are available of: the problems with the prior ontology, e.g., a logical contradiction between the predictive theory and some empirical evidence; the new ontology with the fault resolved; and, an account of the reasoning which led to it.

To demonstrate and help evaluate the ability of ontology evolution, we implement GALILEO within Isabelle (Paulson, 1994), a generic higher-order theorem prover, and use the Isar language (Wenzel, 2007), a human-readable and machine-checkable language for writing proofs. Isabelle provides substantial support for polymorphism and higher-order reasoning, which are vital to our work. Essentially, GALILEO can be seen as an extension to Isabelle, providing the additional functionality of evolving higher-order ontologies.

1.3 Results

In this thesis, we present a collection of ontological conflicts that have occurred over the development of Physics. We have formalised each of these case studies in higher-order logic and implemented each as part of an Isabelle theory. The case studies were split into development and test sets, so that the evaluation of the ontology repair plans is independent from the development. We have implemented the ability to correctly diagnose and repair all of these in GALILEO, such that either the historical solution or solutions representing plausible alternative theories that can be of Physics interest can be found in the search space. Because we work with higher-order, polymorphic terms, the number of unifiers found by higher-order unification is huge. We have evaluated the effectiveness of our technique for pruning the search space to a manageable size. For each solution proposed by the system, we discuss whether it could potentially be an interesting alternate theory, i.e. a repair that was not accepted by the physicists at the time but could potentially give meaningful Physics. Note that the formalisation of Physics is intended to be an idealisation or simplification, because we are not intending to produce a complete formalisation of all of Physics, but only small fragments as needed for the development and evaluation of GALILEO. Details of the results are presented in §8.

1.4 Research Hypotheses

The aim of this project is to demonstrate that automated ontology evolution via ontology repair plans is computationally feasible and can account for the kinds of ontology evolution that are observed in human problem solving in Physics. The main hypothesis that will be evaluated in the project is that:

> *A few generic, ontology repair plans can account for a large number of historical instances of ontology evolution in the Physics domain.*

The evaluation of the repair plans will assess to what extent they create a new ontology that escapes the failures diagnosed in the prior ontology and to what extent this emulates the historical process of ontology evolution.

The power of ontology repair plans largely depends on the logic underlying the formalisation and their implementation in GALILEO. The limited expressivity of first-order logic, let alone fragments of it such as description logics (DL), constitutes a limit on the modelling of both the examples of ontology evolution and of the ontology evolution process. Without the means to quantify over and to reason about the functions, it is virtually impossible to formalise and automate *sufficiently generic* ontology evolution procedures. A key objective is, therefore, to study automated ontology evolution in higher-order logic (HOL). HOL yields various benefits but also challenges that are current topics of interest to the automated reasoning community.

Employing a logic as expressive as HOL for ontology evolution inevitably brings numerous additional technical challenges. The choice required striking a compromise between the expressiveness of the representation and the efficiency of the reasoning process. In order not to burden ourselves with knowledge encoding problems, we have favoured representational richness. The price we have paid is that inference must often be interactive. When ontology evolution is much better understood, it will be time to increase the degree of automation by exploring the potential for representation of Physics and ontology evolution in a more restricted logic.

In order to demonstrate that our techniques are feasible under practical settings, we must show that GALILEO has incorporated mechanisms for addressing some key technical challenges of evolving ontologies in HOL. A subordinate hypothesis that will be evaluated is that:

> *A few heuristics enable: (i) substantial control over the size of the search in the space containing solution candidates, which is otherwise unmanageable, and (ii) preservation of only physically relevant solutions.*

All of these hypotheses have been successfully evaluated, as discussed further in Chapter 8.

1.5 Contributions

In summary, the main contributions of this thesis are:

- The introduction and analysis of a novel approach to automating ontology evolution and evolving higher-order logic ontologies.
- A collection of formal mechanisms, which we call *ontology repair plans*, where each is designed to be triggered upon the detection of a certain type of ontological fault and to resolve the detected fault.
- A system that mechanises ontology evolution and integrates theorem proving into the ontology evolution process, ensuring the soundness of the repaired ontologies.
- A novel approach to the automatic diagnosis of ontological faults by returning physically meaningful candidates of repairs to the ontologies.
- Formalisations of a wide range of case studies from the domain of Physics that are examples of ontology evolution. We have used these case studies to evaluate our system.

1.6 Plan of Thesis

Chapter 2 reviews the literature related to ontology evolution in Physics, covering various areas including general knowledge-base evolution and scientific discovery, and relates our work to the field. Chapter 3 outlines the background of the project, including the knowledge central to the design and implementation of GALILEO and clarifies some of the key notions and terminologies used throughout the thesis with our intended meanings. Chapter 4 begins the investigation with a description of our ontology repair plans, each of which is described with a motivating example and a formalisation. Chapter 5 sets out an overview of the GALILEO system and the modelling approach adopted to formalise the case studies; we present the architecture of the system, introduce the key components, and describe the flow of the system. Chapter 6 considers the implementation of conflict diagnosis and visits some of the research objectives specific to the implementation of GALILEO and the way limitations can be minimised. Chapter 7 provides a description of the mechanisation of ontology repair. Chapter 8 provides an evaluation of both the ontology repair plans and the GALILEO system, and examines each of the research objectives. Chapter 9 discusses avenues for further research and directions in which the scope of the research could be expanded, and draws together the conclusions and summarises the thesis.

Chapter 2

Literature Survey

2.1 Introduction

Although relatively little previous research effort has been directed towards *automated* ontology evolution, this section outlines a rich literature related to its aim and the general methodology. §2.2 presents the background behind agent technology and automated negotiation in multi-agent systems, including an overview of several popular meta-representation languages and a discussion of their semantics. For a review of prominent ideas dealing with the related semantics and reasoning, §2.3 focuses on non-monotonic reasoning, including belief revision. §2.4 presents a survey of research in ontology dynamics, covering a range of techniques used for seeking agreement between semantically incompatible ontologies; these include research in ontology mapping, morphism, matching, debugging and evolution. Finally, §2.5 provides an overview of (semi-) automated scientific reasoning by summarising several successful scientific and theorem discovery systems.

2.2 Ontologies and Multi-Agents

Potential problems that arise from communication among heterogeneous agents in an open environment have stimulated studies of appropriate software paradigms. The most widely recognised and accepted paradigm is the implementation of multi-agent systems, or simply *agents* (Wooldridge, 2002). Interest in the field has been growing rapidly over the last decade, due to the increased emergence of applications involving vast distributed systems.

2.2.1 Ontology Languages

The most fundamental ontology research and development is on the language used for authoring ontologies. The ontology language sets the degree of expressiveness and tractability of the representation. Although the overall development is still at a relatively early stage, ontology languages have significantly evolved to provide better support for ontology integration and interoperability.

Languages based on a markup scheme, e.g., XML, have been recommended as standard languages for the Semantic Web. Since the beginning, the standard language has been constantly evolving – from RDF to OWL. The *Resource Description Framework* (RDF) (Manola et al., 2004) is extended from XML, such that relationships between resources, i.e. subjects, predicates and objects, are expressed as triples. Such representation allows easier extraction of the subjects and objects in question, without the need to refer to a schema. However, RDF does not offer mechanisms for describing the attributes of, or the relationships between, resources. RDF Schema (RDFS) (Brickley and Guha, 2004), a semantic extension of RDF, is a language for describing RDF vocabulary and structuring resources in an object-oriented manner. RDFS is still limited in various aspects, e.g., it cannot express useful constraints on predicates, such as those concerning cardinality and existence. For instance, one cannot express that prime numbers are natural numbers that have exactly *two* divisors. The *Web Ontology Language* (OWL) (W3C, 2012) supports such constraints, uses RDF syntax, and offers three variants, two of which are based upon description logics (Baader et al., 2003).

Description Logic (DL) is a knowledge representation paradigm for representing concepts and their hierarchies. Collections of objects are called *concepts* in DL[1], which are interpreted as sets. Properties and relationships between concepts are represented as binary predicates called *roles*, e.g. $has\_child(x,y)$. DL knowledge-bases comprise two components: TBox, which introduces terminology and defines the vocabulary, and ABox, which contains ground sentences, assertions over the terms in the TBox. Thus, structural relationships between concepts and roles are specified in the TBox, whilst relationships between concepts and individuals w.r.t roles are specified in the ABox. DL implements *open-world semantics*, under which, as long as a fact is not provable from the knowledgebase, it is assumed to be unknown. Conventional relational databases employ the *closed-world semantics*, which assumes the knowledge-base contains all relevant facts and, if a fact cannot be derived, it is assumed to be false. In contrast,

[1]Collections of objects are called *classes* in OWL.

it makes more sense to use open-world semantics for representing ontologies in the Semantic Web, due to the enormous and partially capturable World Wide Web. It is therefore accepted by the community that a deductive reasoner for the Web should not assume a statement to be true or false on the basis of a failure to *disprove* or *prove* it. Open-world and closed-world semantics are discussed in more details in §2.3.

2.2.2 Ontologies in Multi-Agents

The realisation of the Semantic Web vision relies heavily on the success of agent technology (Hendler, 2001; Shadbolt et al., 2006). Depending on the type of problem being confronted, an agent should satisfy a set of properties that define its basic characteristics. For many researchers, *autonomy* is a definitional prerequisite and is useful in distinguishing agents from other types of intelligent software. This provides agents with the ability to "act without direct intervention from humans and have some control over the agent's own actions and internal states" (Castelfranchi, 1995). Agents should also exhibit several other characteristics, including *social ability*, *reactivity*, and *proactivity*. By these characteristics, an agent is expected to be able to interact with other entities, perceive and analyse its environment, and exhibit goal-directed behaviour by taking the initiative (Wooldridge and Jennings, 1995).

The introduction of agents has spurred research into ontologies: into their representation, engineering and use in reasoning. To interact with the environment, agents rely on ontologies which allow them to function by using the available concepts and relationships among them. Unfortunately, in a distributed environment, it is often impractical for agents to access a shared ontology due to its potentially poor maintainability, extensibility and scalability. If a shared ontology is used, then the vocabulary of the ontology needs to be hard-coded in each of the participating agent. This limits flexibility because an update to the existing ontology requires an update to the vocabulary in every agent as well. Moreover, it is extremely laborious to represent the knowledge about a considerable part of the world, given the underlying inherent complexity, and organise it into a few ontologies; one example is the building of the CYC knowledge base (Lenat, 1995; Witbrock, 2011). So, instead, agents need access to multiple ontologies, including giving one or more internal ontologies to each agent, in order to better capture various views of the world; as such, this imposes a risk of failure in agent communication due to disagreement over the representation of concepts across the different ontologies.

2.2.3 Automated Negotiation

Much research has been focused on automating multi-agent negotiation, which is a form of interaction for agents to undertake a kind of collaborative problem solving. Typically, the kind of problem addressed deals with determining possible future courses of actions or future states (Atkinson et al., 2005; Dunne and Bench-Capon, 2006). Studies in this area cover a wide range of interaction and decision mechanisms; for instance, game-theoretic approaches (Rosenschein and Zlotkin, 1994; Binmore and Vulkan, 1999); heuristic-based approaches (Faratin et al., 2002; Rahwan et al., 2007); and argumentation-based approaches (Kraus et al., 1998; Kakas and Moraitis, 2006; Amgoud et al., 2007). Game-theoretic approaches are heavily based on game theory (Osborne and Rubinstein, 1994), a branch of economics that studies strategic interactions between individuals. However, a major limitation of classical game-theoretic approaches is that the space of outcomes must be pre-determined (Parsons and Wooldridge, 2002). It is unrealistic to make such assumption in dynamic environments, as values in these environments are mostly variables so such assumption conflicts with the very nature of these environments. Heuristic-based approaches rely on a set of rules that are typically designed offline; these rules provide approximations to the rational decisions offered by game-theoretic approaches. However, heuristic-based approaches often choose outcomes that are sub-optimal, as they do not cover the entire space of possible outcomes and the adopted notion of rationality is only approximated (Jennings et al., 2001). Argumentation-based approaches are of most relevance to our project, as they are typically highly adaptive which, therefore, renders them the most suitable approaches for working in dynamic environments. More importantly, argumentative reasoning is utilised to allow additional information to be exchanged, which may include the explicit opinion of the agent. With this capability, an agent can offer a critique of the proposal received and explain why it is unacceptable or how it could be modified to become acceptable. For ontology evolution negotiation, an agent must be able to expressively describe the current fault, if there is one, in the agent's own reasoning, and argumentation-based approaches provide the basis for implementing a suitable framework.

Despite a vast amount of research in argumentation-based negotiation in multi-agent systems, very few have addressed the problem of ontology mapping or matching by negotiation. Bailin and Truszkowski (2002) have introduced a protocol for discovering ontology conflicts and negotiating the mismatches between each agent's ontology. The negotiation process is comprised of four major tasks: interpretation, clarification, relevance evaluation and ontology evolution. Interpretation is for determining whether the received message has been correctly understood; clarification is for requesting further information in order to correctly understand the received message; relevance evaluation is for measuring how well the result matches with the expectation; and, ontology evolution is for modifying the ontology of the agent by introducing a new concept, a new representation of an existing concept, or some new constraints. This protocol can be powerful because it imitates negotiation between individuals in the real world. However, a limitation of this approach is that the protocol must be hard-coded in each of the participating agents. To relax the constraint of hard-coding a specific protocol in participating agents, one proposed method is to utilise a shared ontology of negotiation, which contains the vocabulary that agents use for the negotiation session (Tamma et al., 2002). The negotiation protocol regulating interactions is advertised when an agent participates in a pre-existing interaction. An advantage of this approach is that the agents do not need to commit to a particular negotiation protocol, increasing agents' ability to adapt to new environments. However, a shortcoming is that the agents must be regulated by a shared ontology, reducing the autonomy of the agents. Also, it is typically undesirable and infeasible to enforce the adoption of a shared ontology in an open environment such as the Internet.

2.3 Non-monotonic Reasoning

Incomplete information prevails in our everyday reasoning as it is unrealistic to assume that conclusions can be drawn based on *all* relevant information. Further, it is unreasonable to assume that knowledge is static and that updates are not required. Classical logic systems typically are monotonic, i.e. if a sentence p follows from a set of propositions A, then p also follows from a set B, where $A \subset B$. Monotonicity does not allow the system to make inferences which may later be retracted when further information is added. Given two premises "birds fly" and "Tweety is a bird", an agent can infer that "Tweety flies". In the light of additional information that "Tweety is a kiwi", monotonic systems cannot retract theorems even when they should.

2.3.1 Closed-World Assumption, Default Logic, and Circumscription

The closed-world assumption (CWA), proposed by Reiter (Reiter, 1978), is the earliest well known non-monotonic reasoning scheme. The idea is to assume false all facts not provable to be true, so a system employing the CWA does not require explicit representations of negative facts in order to derive negative inferences. Being the basis of database theory, a flight database, for instance, only needs to contain details of all known flights to answer queries concerning both existing and non-existing flights, because queries against unknown flights automatically give negative results. Clearly, the expressivity of CWA is very limited, as the underlying assumption is very strict, in the sense that failure in *provability* determines the *truth* of the proposition.

A more flexible formalism is Reiter's *Default Logic* (Reiter, 1980), which allows variables to be assigned values under "normal circumstances". Default statements, or defaults, are treated as inference rules, rather than formulae. Defaults are in the form $\frac{\varphi:M\psi}{\psi}$, which captures the intuition that if the *prerequisite*, φ, holds and the *justification*, $M\psi$, can be consistently assumed, then the *conclusion*, ψ, can be inferred. In the Tweety example, $M\psi$ is inconsistent with what is known because kiwis cannot fly. The set of defaults for the Tweety example may contain, e.g., the default $\frac{bird(x):Mfly(x)}{fly(x)}$. Defaults can be paired with a set of known facts containing, e.g., $bird(tweety)$ for expressing that Tweety is a bird, to become a *default theory*.

McCarthy's *circumscription* approach is another way of formalising non-monotonic reasoning; note that it is not a non-monotonic logic but non-monotonic reasoning based upon first-order logic (McCarthy, 1980). The intuition is to pick out all minimal models of a theory, where a model M of a theory T is minimal if there is no other model N of T such that M sets true all variables N sets to true. This is a general formalisation of the CWA, as what is not specified is assumed false. Circumscription and Default Logic differ in several ways: for example, circumscription works with minimal models while Default Logic works with arbitrary models. McCarthy also introduced the concept of an abnormality predicate, denoted as $ab(x)$ (McCarthy, 1986), which can be used to represent that the proposition does not hold in normal circumstances. For instance, for the Tweety example, one could express that birds normally fly with $\forall x.\ bird(x) \wedge \neg ab(x) \rightarrow fly(x)$ (Robinson and Voronkov, 2001).

2.3.2 Belief Revision

An example non-monotonic form of reasoning is the process of changing an agent's beliefs to accommodate new information, possibly inconsistent with existing beliefs. The major focus of research here is to investigate possible models of belief change, known as belief revision operators, and demonstrate that they exhibit properties that resemble intuitive rationality. Rational belief revision operators must adopt reasonable, coherent revision.

Although there are many variations to the definition of rationality, almost all of them incorporate the principle of *minimal change*, which governs the need to preserve as much of earlier beliefs as possible. Unfortunately both in philosophy and in artificial intelligence, there is no single answer to achieving minimal change. The best known attempt to characterise minimal change is the AGM model (Alchourrón et al., 1985), which has considered three forms of belief change: expansion, for adding a new belief to the belief set without regard to consistency; revision, for adding a new belief to the belief set and removing other beliefs to maintain consistency; and contraction, for removing a belief from the belief set. The postulates for a single-round of revision, which is the most trivial belief revision strategy, are as follows:

(R1) $K * \alpha$ is a belief set. *Revising K with* α *gives a belief set.*

(R2) $\alpha \in K * \alpha$. *Revising K with* α *gives a set containing* α.

(R3) $K * \alpha \subseteq Closure(K \cup \{\alpha\})$. *The belief set revised with a new belief contains only beliefs implied by the combinations of the old beliefs with the new belief.*

(R4) If $\neg\alpha \notin K$, then $Closure(K \cup \{\alpha\}) \subseteq K * \alpha$. *If the new belief is consistent with K, then the beliefs implied by the combination of the old beliefs and the new belief make up the revised belief set.*

(R5) $K * \alpha = Closure(false)$ if and only if $\vdash \neg\alpha$. *The revised belief set is inconsistent if and only if the new belief is inconsistent.*

(R6) If $\alpha \leftrightarrow \beta$, then $K * \alpha = K * \beta$ *The revised revision process abides by the principle of Irrelevance of Syntax, i.e. not to be affected by the syntactical forms of the new belief.*

(R7) $K * (\alpha \wedge \beta) \subseteq Closure((K * \alpha) \cup \{\beta\})$. *The belief set revised with* $\alpha \wedge \beta$ *contains only beliefs implied by the combination of the beliefs revised with* α

and β.

(R8) If $\neg\beta \notin K * \alpha$ then $Closure((K * \alpha) \cup \{\beta\}) \subseteq K * (\alpha \wedge \beta)$. *If* β *is consistent with the belief set revised with* α*, then the beliefs implied by the combination of the beliefs revised with* α*.*

where K is a belief set and α and β are formulae; $*$ is a revision operator; and, $Closure(K)$ returns the deductive closure of K. In (Katsuno and Mendelzon, 1991), Katsuno and Mendelzon revised the AGM postulates to restrict a state of belief to a propositional formula. A revision operator $*$ on knowledge sets (theories) satisfies R1 to R6 iff a corresponding revision operator $\circ$ on beliefs satisfies KM1 to KM4:

(KM1) $\psi \circ \mu$ implies μ [$\equiv$ R2]

(KM2) If $\psi \wedge \mu$ is satisfiable, then $\psi \circ \mu \equiv \psi \wedge \mu$ [$\equiv$ R3 and R4]

(KM3) If μ is satisfiable, then $\psi \circ \mu$ is also satisfiable [$\equiv$ R5]

(KM4) If $\psi_1 \equiv \psi_2$ and $\mu_1 \equiv \mu_2$, then $\psi_1 \circ \mu_1 \equiv \psi_2 \circ \mu_2$ [$\equiv$ R6],

where ψ is a propositional formula representing a knowledge-base and μ is a sentence respectively.

The semantic model of the AGM postulates is a partial pre-order of interpretations for expressing the plausibility, or the *epistemic entrenchment*, of interpretations (Gärdenfors and Makinson, 1988). The interpretation with the highest degree of epistemic entrenchment is given the highest rank and is designed to be the most preferred. Ideally, the most epistemically entrenched interpretation is the one that is the most appropriate after revising the knowledge-base with the new information, while maintaining rationality in the process. However, there is no unique solution to the assignment of ranks.

Since the AGM theory is designed for mere one-step revision, each application of a revision operator satisfying the postulates is independent of the others, i.e. the current revision is not influenced by the results of previous revisions. Thus, the AGM postulates are too weak for handling *iterated revision*, which is the process of revising beliefs with new information iteratively. Iterative belief revision is closely connected to *conditional beliefs*, i.e. beliefs depending on earlier beliefs. Darwiche and Pearl (Darwiche and Pearl, 1997) have proposed additional postulates which are designed for preserving the coherence of both conditional and unconditional beliefs, and for achieving absolute minimal change in conditional beliefs. Similarly, Freund and Lehmann (2002) have described an additional postulate ensuring that if the new information is inconsistent, none of the previous beliefs is to be retained. Many other postulates have been proposed, but all behave well only in a specific problem setting.

None of the work in this area considers syntactical changes, so the signature of the world is assumed to be static. This assumption contradicts that underpinning ontology evolution, because the conceptualisation of the world is expected to change; therefore, the signature of an ontology is also expected to be updated by an agent. Furthermore, although the AGM postulates formalise the set-theoretic properties of the revision process, belief revision techniques are typically studied by assuming the underlying language is propositional logic. In contrast, the complex descriptions of Physics objects and properties will require a much more expressive language, e.g., higher-order logic. However, the philosophical issues that belief revision confronts are somewhat similar to those faced by ontology evolution, e.g., rationality and minimal change.

2.4 Ontology Dynamics

As the world inevitably evolves over time, it is unrealistic to assume static knowledge and information. When working with knowledge-bases that capture the existence of entities in the world or the relationships among these entities, one has to deal with disagreements over different conceptualisations. Conflicting conceptualisations are simply results of incompatibilities between the underlying assumptions made about the domain. When a domain changes, subsequent updates performed to the knowledge-base essentially replace the original assumptions made about the domain with the new. Even when the domain is fixed, there are situations in which the assumptions made about the domain could still be incompatible; for instance, the selection of different interpretations or perspectives of the domain is necessarily based on unequal sets of underlying assumptions. Regulating the assumptions made about a domain is almost impossible to achieve in an open, distributed environment.

2.4.1 Problem Background

Even in fields that inherently allow more control over the data, such as traditional database systems, maintaining data integrity is a long standing challenge (Davenport, 1976; Baltopoulos et al., 2011). Data integrity can be compromised in a number of ways, including a change in the schema of a table. Designing mechanisms for handling schema evolution represents an unsolved problem for information systems that is further exacerbated in distributed information systems, such as online scientific databases; various attempts have been made in this direction (Ventrone, 1991; Roddick, 1992; McBrien and Poulovassilis, 2002; Curino et al., 2008).

The problem of ontology change is even more complex than data integration. Ontologies describe a domain of discourse, where concepts and relations have formally defined semantics for machines to interpret. Moreover, ontologies themselves incorporate semantics, such that the definitions are essentially sets of logical axioms. Thus, ontology data models are rich, as there are a large number of possible representation primitives available, depending on the expressiveness of the ontology language. Even with some common languages such as those based on description logics, the primitives are not merely inserting/removing instances from a class or introducing new superclasses, but defining classes as unions or intersections of other classes, redefining cardinality constraints, and so forth. When an expressive logic is adopted, such as first-order or higher-order logic, the representation primitives become even more complex, e.g., increasing the arity of a function, altering the type of a symbol, and so forth. There are various approaches to resolving ontological heterogeneity and the key approaches that are most relevant to our project can be summarised as: mapping, morphism, and matching (Flouris et al., 2008).

2.4.2 Ontology Mapping

The goal of ontology mapping is to construct mappings between concepts across some given ontologies. According to Kalfoglou and Schorlemmer (2003b), *ontology mapping* refers to

> *the task of relating the vocabulary of two ontologies that share the same domain of discourse in such a way that the mathematical structure of ontological signatures and their intended interpretations, as specified by the ontological axioms, are respected.*

By this definition, mappings are created only between signature symbols and not between axioms. Even though ontology mapping only relates signature symbols, it is already difficult, as the development of techniques is restricted by several inherent methodological limitations; for instance (Kalfoglou and Schorlemmer, 2002):

- The assumptions underlying and the justifications supporting the creation of the mappings are often not made clear or exposed to the community;

- Many implementations of ontology mapping are integrated in ontology editing environments or are attached to a specific formalism without including the specifics of the approach to devising mappings;
- The semantics of concepts are often neglected when devising mappings, so many proposed techniques are only syntactic;
- There is no single definition of the term *ontology mapping*, so it has different meanings in different works, which creates unnecessary confusion among those attempting to formalise the theory.

Most simpler approaches compute “similarity” coefficients based on syntactic features such as concept labels, which usually yield limited accuracies because syntactic features are descriptively inadequate for capturing the semantics of ontological concepts. More accurate but complex approaches attempt to incorporate mechanisms for dealing with semantics: for instance, by computing the semantic relations between concepts (Giunchiglia et al., 2004; Guinchiglia and Yatskevich, 2004; Giunchiglia et al., 2007) and by discovering missing background information from the web (Gligorov et al., 2007) or ontologies on the Semantic Web (Sabou et al., 2008). S-Match (Giunchiglia et al., 2004; Guinchiglia and Yatskevich, 2004; Giunchiglia et al., 2007) implements the notion *semantic matching*, which computes the semantic relations between two concepts. It views an ontology as a taxonomy represented in a graph, so it analyses the structure and determines the semantic relations depending on the position of a concept in the graph. Gligorov et al. (2007) have made use of an Internet search engine to help create mappings between concepts even in inherently imprecise domains such as Music genres. It relies on the notion of *approximate ontology matching* between concepts, which utilises a weighting function for evaluating the appropriateness of the mappings.

2.4.3 Ontology Morphism

The process of *ontology morphism*, where the term *morphism* comes from Category Theory, is the construction of functions that relate both the signature symbols and the axioms of the ontologies (Kalfoglou and Schorlemmer, 2003b). Morphisms are structure-preserving mappings between two mathematical structures, so they are more ambitious than mappings. The concept of ontology morphism naturally comes about when ontologies are formalised with an algebraic approach and draw parallels with signature and theory morphisms (Bench-Capon and Malcolm, 1999). Kalfoglou and

Schorlemmer (2003a) formally relate ontology morphisms to the notions of logic infomorphisms stemming from Information-Flow theory (Barwise and Seligman, 1997).

2.4.4 Ontology Matching

In both ontology mapping and ontology morphism, ontologies are related by constructing functions. A relatively more popular approach for determining correspondences between ontologies is *ontology matching*, which devises relationships between ontologies instead. It gives greater flexibility than ontology mapping and morphism, because such relationships capture more complex semantics including subsumption, equivalence, disjointness, and any user-specific notion of similarity (Kalfoglou and Schorlemmer, 2004). Most matching techniques focus on the computation of some kind of semantic distances between concepts in order to establish relationships between ontologies. A widely employed tool for ontology matching is WordNet (Miller, 1995), which is an electronic lexical database where the different senses of words are grouped together into sets of synonyms; it is often used for finding synonyms and related terms. Most common measures of similarity are based on some kind of numerical degree $s \in [0,1]$ (Ichise, 2008; Formica, 2008), so machine learning and probabilistic techniques are popular methods to improve the quality of ontology alignments. For instance, (Doan et al., 2004) exploits information in the taxonomical structure of the ontology and utilises a probabilistic model to combine the results of a set of learners to find relationships. For more complex matching, Giunchiglia and Shvaiko (2004) have described an approach that grounds the concepts in an ontology in WordNet terms and formulates the task to become a constraint-satisfaction problem, whilst Niepert et al. (2010) have presented a Markov logic based probabilistic-logical framework.

2.4.5 Ontology Debugging for Description Logics

When an internal error is realised in an ontology, techniques such as ontology mapping, morphism, and matching are insufficient to resolve the problem. Those techniques address the problem of finding semantic relationships between the concepts, relations and axioms in multiple ontologies, and not to make persistent changes to the signature or the axioms. The general task performed by those techniques can be viewed as adding new information to the meta-level, because the resulting mappings represent relationships between ontologies, which does not resolve internal faults in each ontology itself; object-level modifications are required in order to update the signature and/or the axioms.

For DL ontologies, ontological faults come in two forms (Haase and Qi, 2007): An ontology is

- *inconsistent* if and only if it has no model
- *incoherent* if and only if there are unsatisfiable concepts in its TBox which are interpreted as empty sets.

Kalyanpur et al. (2006b) have described an approach for repairing incoherent OWL ontologies by identifying erroneous axioms. This is done by computing *Minimal Unsatisfiability Preserving Sub-TBoxes* (MUPS). A MUPS for a concept C is a minimal fragment of the knowledge-base in which C is unsatisfiable. The technique involves ranking the axioms in each of the MUPS generated according to a range of attributes, including axiom frequency and the impact on the ontology if the axiom is removed; the axiom with the lowest rank is removed from the ontology. This approach has been extended to repair inconsistent ontologies as well (Kalyanpur et al., 2006a). The extension involves discovering the erroneous *part* of axioms by splitting the original axioms into smaller parts and then apply almost the same method for repairing incoherent ontologies. Ji et al. (2009) have proposed a method for debugging both incoherent and inconsistent ontologies as well. For incoherent ontologies, MUPS are calculated for debugging incoherence and *minimal inconsistent subsets* (MIS), which are minimal subsets of the ontology that cause it to be inconsistent, are calculated for debugging inconsistency. Typically, inconsistency is resolved by removing one or more axioms from all generated MIS. Ribeiro and Wassermann (2009) addresses ontology debugging with a belief revision approach (§2.3.2) and introduces new operators that can be used in DL.

Haase et al. (2005) compares four main types of approaches for handling inconsistencies in dynamic ontologies, which are ontologies that depend on changing parameters such as time, space, environment, etc. These approaches – namely, consistent ontology evolution, repairing inconsistencies, reasoning with inconsistent ontologies, and ontology versioning – are conventionally recognised as different approaches to the same problem. The comparison revealed that they, in fact, have very different requirements and are applied in different settings; for example, inconsistency repair removes logical inconsistency from the input ontology during development-time, whilst inconsistency reasoning focuses on evaluating the meaningfulness in answers during runtime. Moreover, repairing ontological inconsistencies involves locating the inconsistencies and then resolving them, whereas the operations adopted to keep resulting ontologies consistent resemble those in belief revision and contraction (Gärdenfors and Rott, 1995).

2.4.6 Engineering OWL ontologies

Theoretical issues, such as deciding whether every concept satisfiable relative to one TBox $\mathcal{T}_1$ is also a concept satisfiable relative to another TBox $\mathcal{T}_2$, i.e. whether $\mathcal{T}_2$ is a *conservative extension* of $\mathcal{T}_1$[2], have been argued to have an impact on the quality of modified ontologies. A detailed study of the desirability of the notion of conservative extensions in DL is presented by Ghilardi et al. (2006), which describes algorithms for deciding *relativised* and *non-relativised* conservative extensions of ontologies in DLs. The notion of relativised conservative extensions is particularly useful to ontology evolution because modified ontologies are often not conservative extensions of a TBox w.r.t. the entire signature of the TBox, but only to a subset of it.

Another central issue is how to improve reusability of (parts of) ontologies by extracting meaningful fragments, or *modularisation*. A module of an ontology has been defined as:

> ... *a reusable component of a larger or more complex ontology, which is self-contained but bears a definite relationship to other ontology modules* (Doran, 2006),

so the purpose of ontology modules is to allow them to be reused as they are or extended with new concepts and relationships. Enhancing the reusability of ontologies is an important issue because practical ontologies are usually very large, e.g., the medical ontology SNOMED-CT (Stearns et al., 2001) has over 360,000 unique concepts and 1.2 million relations. Importing the entire ontology is, therefore, infeasible and excessive if one would like to reuse only a fragment of the original ontology for constructing a new ontology. The notion of a module is in fact closely related to the notion of a conservative extension, but different in an important way. With a conservative extension, the logical consequences are considered only w.r.t. to the ontologies of interest, whereas a module considers all possible ontologies in which the module can be used

[2]This is a restricted definition of conservative extensions, commonly called *non-relativised* conservative extensions. Generally, *relativised* conservative extensions are more commonly investigated in research, i.e. Γ-conservative extensions where $\Gamma \subseteq Sig(\mathcal{T}_1)$.

(Grau et al., 2007). Grau et al. (2001) have also formally proved that the notions of modules and conservative extensions are in fact closely related from both proof- and model-theoretic views. As the problem of deciding whether an ontology is a conservative extension of another is already undecidable for the OWL DL, the problem about modules can only be more complex because the definition of a module is more restrictive than that of a conservative extension. Thus, the extraction of modules is typically done by approximations.

2.4.7 Interactive Ontology Evolution

Most research activities in ontology evolution are led by interactive ontology evolution, which requires users to explicitly guide the evolution process. Thus, most of the current work on ontology evolution aims to help users edit ontologies manually by providing informative feedback.

One of the pioneering works in interactive ontology evolution is by Stojanovic et al. (2002), where a specialised cyclic process is introduced for ontology evolution. It systematically verifies the consequences of the major operations, including *change implementation* and *change propagation*. The change implementation phase involves integrating the proposed changes to the given ontologies, whilst the change propagation phase allocates induced changes to all affected ontologies. Some recent works have looked at the computation of explanations for unsatisfiable concepts and inconsistent ontologies, or *justifications* (Kalyanpur, 2006). These are minimal subsets of an ontology that are sufficient for a given entailment to hold. Inspired by users' interests in the automatic explanation function provided by Swoop (Kalyanpur et al., 2006c), Kalyanpur et al. have presented both reasoner-dependent (glass-box) and -independent (black-box) algorithms for computing justifications of an entailment in OWL DL (Kalyanpur et al., 2007).

2.4.8 Progress in Automated Ontology Evolution

Progress in automated ontology evolution has been slow due to the scale and complexity of the problem. One example system in this area is the *Ontology Repair System* (ORS) (McNeill and Bundy, 2007), which addresses the problem of ontology repair in a multi-agent planning environment. It was designed to investigate an environment in which agents with slightly different ontologies interact with each other in order to offer or seek services to and from others. The main goal of ORS is to identify and repair ontological mismatches arising from heterogeneity in the underlying logical representation. The type of ontology *repair* implemented differs from the more common ontology *mapping* or ontology *matching* in being aimed at identifying and correcting errors in a single ontology rather than constructing a mapping or alignment between two ontologies. Ontology repair is also done automatically, dynamically (at run-time) and without full access to the ontologies of other agents, whereas typically ontology matching has some manual element, is done statically (at compile-time) and with full access to both ontologies.

The ORS ontology repair operations support various syntactic manipulations that are beyond belief revision, including changing the arities or order of arguments of a function and splitting a function into parts. Such manipulations are similar to those developed in this project; however, we have also investigated more complex conceptual faults commonly occurring in problem solving.

Evolva is another similar attempt at automating ontology evolution, but its apparent focus is more toward the management of ontological knowledge rather than the resolution of conflicts between ontologies (Zablith, 2008). It exploits background information for the integration of new external information to given ontologies, which aims to reduce, or even eliminate, the need for user input. Sources of such background knowledge include WordNet and web corpora. Only vague details on the handling of ontological inconsistencies have been revealed, which is likely due to the preliminary state of the current work.

2.5 Scientific Problem Solving

As widely agreed by AI researchers, intelligence involves creativity. Progress towards more intelligent machines has motivated attempts at tackling problems known to be seemingly solvable only by human creativity (Rowe and Partridge, 1993). A prime example problem is scientific and mathematical discovery, which requires the researcher to apply creative reasoning in order to invent new theories (Langley, 1998). Due to the enormous search space, most work in this area focuses on formalising human reasoning abilities using heuristics.

2.5.1 Scientific & Mathematical Discovery

One of the earliest heuristic-based machine discovery systems is Lenat's Automated Mathematician (AM) program. It discovers mathematical concepts by performing concept formation and conjecture making in number theory (Lenat, 1977). It starts with about 110 elementary concepts, where each concept is represented by a set of slots. Each slot contains information regarding, e.g., definition, examples, generalisations/specialisations and worth. To fill these initially blank slots, AM looks through a database containing about 250 heuristic rules. Four types of heuristics were used: fill rules, for filling the slots; check rules, for validating the entries; suggest rules, for generating new concepts and new tasks; and interest rules, for measuring the interestingness of the concept. It is important to note that AM is designed to be interactive and the user has a crucial role in shaping the search process. The user can, for example, judge the interestingness of the theorems formulated so far. Some interesting results include the discovery of addition, multiplication, primes, and Goldbach's conjecture.

Another system guided by heuristics is found in the BACON series of programs (Langley, 1980). These programs are purely data-driven, i.e. they generate theories from empirical data only. The heuristics help formulate regularities, such as constancies, trends, common divisors, and constant differences, hidden in the input data. One heuristic is that if the values of one variable increase as those of another variable increase, then their ratio should be considered. This heuristic was used to deduce Kepler's Third law: the data about planetary coordinates and trajectories show that the distance D increased with the orbital period P, but the two are not linearly related, so BACON first defines $\frac{D}{P}$ as the ratio of D and P. It then defines $\frac{D^2}{P}$ as the product of $\frac{D}{P}$ and D because they are inversely related. Ultimately, it defines $\frac{D^3}{P^2}$ as the product of $\frac{D^3}{P^2}$ as the ratio of $\frac{D^2}{P}$ and $\frac{D}{P}$. These heuristics, therefore, can be seen as *ad-hoc* patterns for curve-fitting.

Pease et al. (2004) have used ideas from Lakatos to evolve mathematical theories. For instance, they have automated the repair of faulty conjectures using computational versions of Lakatos's *Proofs and Refutations* methods (Lakatos, 1976). The proposed methods for fixing faulty conjectures in the light of counterexamples include: *monster-barring*, for modifying the definition to exclude unwanted counterexamples; *piecemeal exclusion*, for restricting the conjecture to those examples that do not exhibit properties of the counterexample; and *strategic withdrawal*, for restricting the conjecture to those examples for which it is known to hold. Their approach is contrasted with that proposed here by focusing on adding and changing definitions and conjectures within a signature that is changed only by definitional extensions. Our work also employs definitional extension, but is principally focused on more radical signature changes.

The GALILEO system is also guided by heuristics, which formalise the methodology a problem solver adopts to reduce the search space. Both GALILEO and BACON target at Physics domain, but the two systems are strategically different. For example, BACON can be seen as a curve-fitting program, whereas GALILEO attempts to make representational changes in order to repair faults in reasoning.

2.5.2 Qualitative Problem Solving

Motivated by the apparent tendency that humans describe and reason about physical environments qualitatively, various attempts have been made to formalise the underlying qualitative calculus. In the "naive Physics" of Hayes et al. (1978), the description of a situation is more explicit than one in formal Physics by the inclusion of commonsense knowledge that may be taken for granted. Hayes' analysis described the concepts and axioms involved in order to formally understand the behaviour of a liquid (Hayes, 1985).

De Kleer and Brown (1984) have also studied specific physical phenomena using a qualitative modelling approach, including the mechanics and structure of a pressure-regulator. The modelling strategy adheres to several principles that distinguish the approach from other architectures for building theories for specific situations. One principle is *no-function-in-structure*, which forbids the laws of parts of a device to presume the functioning of the whole. Take as an example the model of a switch stating that *if the switch is on, current flows*. The description is false as there are contexts in which current does not flow even if the switch is on, e.g., the circuit is for some reason open even when the switch is on. This principle aims to keep an account of *how* the physical system achieves its behaviour, which is an attempt to reduce the complexity caused by the underlying context of the situation.

Research in designing problem solvers for tackling problems in related areas have been active for over 30 years. The MECHO project (Bundy, 1979) is designed to solve basic high-school mechanics problems stated in English. NEWTON (De Kleer, 1977) is also a solver for mechanics problem but uses qualitative Physics to solve problems, allowing more difficult input problems to be answered. More recent research includes AURA (Chaudhri et al., 2007), which is a knowledge capture system for domain experts to build knowledge bases from university-level science textbook and for a different set of users to ask questions in English against the knowledge base. In AURA, the knowledgebase has been built on top of the *Component Library* (Barker et al., 2001), which is a domain-independent knowledgebase and focuses on the reusability of individual domain-independent knowledge units. Roles, e.g., EMPLOYEE, relate together entities, e.g., PERSON, but do not relate together other roles, so the underlying logic is weaker than first-order. From our research, no all encompassing formulation of Physics exists that we could build on.

2.6 Summary

We have discussed techniques for handling ontology conflicts in multi-agent systems, most of which can be viewed as methods of automated negotiation. We have also described a range of approaches for resolving the general problem of ontology conflicts. These approaches address situations in which a single ontology contains internal faults or multiple ontologies have disagreements with each other. All of these techniques achieve a diversity of specific tasks, but fall short of producing interesting, meaningful matches or repairs. Non-monotonic logics provide formalisms for reasoning about real world situations, where the complete representation of a situation and its underlying contexts can be difficult. Non-monotonic reasoning allows the insertion of a new sentence into the knowledgebase to affect the set of warranted conclusions, but most techniques assume a static language for representing the situation. Scientific problem solving is also an active field of research and numerous systems have focused on tackling various demanding tasks that are significant to the scientific community.

Chapter 3

Background

3.1 Introduction

As described in Chapter 2, the scope of this project is broad and little previous research has been conducted in the area of automated ontology evolution in an expressive logic. That said, the main pillars of the project are the notion of higher-order ontologies and the technique for performing mechanical reasoning in HOL (Chan et al., 2010b). We believe that an accurate description of our notion of ontologies helps specify the technical aspects that provide additional logical power to our representation and clarify any conflicts in terminology used by some others in related fields. Higher-order unification is widely used to mechanise higher-order reasoning, so it is relevant to provide an overview of the key aspects of the process. In this chapter, we present a formalisation of our notion of higher-order ontologies and a polymorphic higher-order logic used to formalise the ORPs designed and case studies for both development and testing.

3.2 Representing Ontologies in HOL

Many ontologies are relatively simple descriptions of concept hierarchies and relations between objects , e.g., taxonomies used for plant classification. A general method for modelling ontologies is to encode a relationship between two objects as a RDF triple. For more expressivity, an ontology can consist of sets of logical formulae. Popular logics used for modelling ontologies include DLs (description logics) and FOL (first-order logic). There are many varieties and sublanguages of DL, e.g., OWL; each involves a compromise between the expressivity of the representation and the efficiency of the reasoning process. We can, thus, see that there is a range of interpretations of "ontology":

- from a single description of the nature of reality to multiple, perhaps conflicting, descriptions of many smaller domains;
- from a basic set of type declarations to a representation of world knowledge; and,
- from a simple classification to a rich logic.

In GALILEO we have interpreted "ontology" in an inclusive way to encompass any representation of knowledge, usually as a logical theory. Thus, meta-ontologies, which are ontologies about other ontologies, are also ontologies. We have chosen a logic which is rich enough to represent, in a natural way, both the object-level concepts and relationships of Physics as well as the meta-level concepts and relationships of the ontology evolution processes. As we will see, this argues for a polymorphic, typed, higher-order logic for both purposes. *Polymorphic* means that a term may have more than one type, which can be achieved by including variables ranging over types in their type declarations. There are two classes of polymorphisms: *parametric* and *ad-hoc* (Strachey, 2000)[1]. Parametric polymorphism occurs when a function is defined over a range of types, acting in the same way for each type. Ad-hoc polymorphism occurs when a function is defined over several types, with potentially different behaviours for each type.

The rest of this section covers the formalisation of our logic and ontologies, most of which are adapted from Lehmann et al. (2012)[2].

3.2.1 Overview of HOL

The formal definitions of our polymorphic, typed, higher-order logic are given below.

Definition 1 (Types) *The following* BNF *describes polymorphic types:*

$$\mathcal{T} \quad ::= \quad \mathcal{V} \mid \mathcal{C} \mid \mathcal{T} \Rightarrow \mathcal{T}$$

where $\mathcal{V}$ is the set of type variables, $\mathcal{C}$ the set of type constants and $\tau \Rightarrow \tau'$ is the type of functions from objects of type τ to objects of type τ'. We will use the Greek letter τ, possibly subscripted or primed, to range over types.

[1] Alternatively, overloading.

[2] The relevant part was produced by Alan Bundy.

Note that the set of type constants, $\mathcal{C}$, depends on the signature (3.2.2) of the ontology. The operator $\Rightarrow$ is right associative.

Particular type constants that we use below are, for instance, *bool*, for the booleans, and $\mathbb{R}$ for the real numbers.

Definition 2 (Order of a Type) *The following equations determine the order of a type:*

$$\begin{aligned} ord(\mathcal{T}) &= 0 \text{ if } \mathcal{T} \text{ is atomic} \\ ord(\mathcal{T} \Rightarrow \mathcal{U}) &= max(1 + ord(\mathcal{T}), ord(\mathcal{U})) \end{aligned}$$

where ord returns the order of a given type.

The order of a type is a relevant concept in this thesis, because we will later discuss specific higher-order functions, e.g., those in §3.3, and it is used by heuristic to prune the search space of repairs.

Definition 3 (Terms) *The following* BNF *describes terms of the logic.*

$$T \quad ::= \quad V \mid C \mid T(T) \mid \lambda V.T$$

where V is the set of term variables, C the set of term constants. We will use the Roman letter t, possibly subscripted or primed, to range over terms. We will write t:τ to declare term t to have type τ.

We will assume that C contains: the truth values $\top$:*bool* (true) and $\bot$:*bool* (false); the standard logic connectives, such as $\wedge$:$bool \Rightarrow bool \Rightarrow bool$; and the quantifiers, such as $\forall$:$(\tau \Rightarrow bool) \Rightarrow bool$, whose meaning can be expressed as

$$\forall(P) \equiv (P = (\lambda x.\top)) \tag{3.1}$$

where P is a predicate.

Formulae are λ-calculus terms of type *bool*. All HOL ontologies also have access to, for each type τ, an equality relation $=_\tau$:$\tau \Rightarrow \tau \Rightarrow bool$, a partial order relation $<_\tau$:$\tau \Rightarrow \tau \Rightarrow bool$ and additive and subtractive functions $+_\tau, -_\tau$:$\tau \Rightarrow \tau \Rightarrow \tau$[3][4]. Note that several of these term constants are *polymorphic*, i.e., their types contain type variables. This allows us to overload these constants so that they apply to objects of many different types. This is essential in allowing us to formalise generic ontology repair plans that are applicable across a wide range of different areas of Physics.

[3]Although these constants are available at all types, Isabelle only has definitions at some types, e.g., naturals, integers and reals.

[4]Note that $-$ can be defined in terms of $+$.

Definition 4 (Type Inheritance) *Terms inherit their types according to the following two rules:*

$$\frac{t'{:}\tau \Rightarrow \tau' \quad t{:}\tau}{t'(t){:}\tau'} \qquad\qquad \frac{x{:}\tau' \quad t{:}\tau}{\lambda x.t{:}\tau' \Rightarrow \tau}$$

Definition 5 (Free Variables) *The set of* free variables, $FV(t)$, *of a term,* t, *are defined recursively as follows:*

$$\forall x \in V.\ FV(x) ::= \{x\}$$
$$\forall t,t' \in T.\ FV(t'(t)) ::= FV(t') \cup FV(t)$$
$$\forall c \in C.\ FV(c) ::= \{\}$$
$$\forall t \in T, x \in V.\ FV(\lambda x.t) ::= FV(t) \setminus \{x\}.$$

The above definitions of types and terms are in, so called, *curried form*, where all functions are regarded as unary. Curried types and terms can be inter-converted as follows:

- The curried type $(\tau_1 \Rightarrow \ldots \Rightarrow (\tau_n \Rightarrow \tau)\ldots)$ can be written in uncurried form as $\tau_1 \times \ldots \times \tau_n \Rightarrow \tau$ and vice versa; and
- The curried term $t(t_1)\ldots(t_n)$ can be written in uncurried form as $t(t_1,\ldots,t_n)$ and vice versa.

We use HOL to refer to an implementation of higher-order logic with a polymorphic type theory hereafter.

3.2.2 Higher-Order Ontologies

We have favoured representational richness and chosen polymorphic, typed higher-order logic for the representation of ontologies, which encompasses all logics commonly used for ontological representations, including DL and FOL.

Definition 6 (Higher-Order Ontologies) *Let* $\mathbb{O}$ *be the meta-type of ontologies. A* higher-order ontology, $O{:}\mathbb{O}$, *is a pair* $\langle S,A \rangle$, *where* S *is the* signature *of* O *and* A *is the set of* axioms. *We define* $Sig(O) ::= S$ *and* $Ax(O) ::= A$.

The signature, S, *is a set of type declarations for a subset of the constants in* C, *i.e., the elements of* S *are of the form* $c{:}\tau$, *where* $c \in C$ *and* $\tau \in \mathcal{T}$. *We will assume that a* *potentially infinite set of variables is provided for each type in* $\mathcal{T}$*. With this assumption, the signature defines a set of terms* T*. These terms are called the* language *of the ontology, denoted* $Lang(O)$*. The set of* sentences *of the ontology,* $Sent(O)$*, are the subset of terms that have type bool and no free variables, i.e.,* $Sent(O) ::= \{\phi \in T \mid \phi\colon bool \wedge FV(\phi) = \{\}\}$ *To reduce clutter, we will sometimes omit outermost universal quantifiers from sentences.*

An axiom*,* $a \in A$*, is a sentence that is assumed to be true. When the axioms are closed under the rules of inference they define a set of* theorems *of the ontology, denoted* $Th(O)$*. If a sentence* ϕ *is a theorem of ontology* O*, we write* $O \vdash \phi$*. If a setence* ϕ *is not a theorem of ontology* O*, we write* $O \nvdash \phi$*, but validity in* HOL *is undecidable.*

An ontology O *is said to be* inconsistent *if* $\bot$ *is a theorem, i.e.,* $O \vdash \bot$*.*

To reduce clutter we adopt the convention that $Sig(O_i)$ is abbreviated to S_i and $Ax(O_i)$ is abbreviated to A_i. Note that, depending on the context, S_i and A_i may refer to the local signatures and axioms, respectively.

$O \vdash \phi$ is an example of a *meta-assertion*, i.e., a sentence of a meta-ontology. A meta-ontology is a higher-order ontology whose domain is other ontologies, i.e., in which the constants are strings denoting object-level sentences, names of ontologies and models, etc. Ontology evolution will consist of simultaneous inference in both object- and meta-ontologies.

An example of a standard, polymorphic axiom inter-relating multiple polymorphic constants is:

$$\forall x, y, z : \tau.\ x +_\tau y =_\tau z \iff z -_\tau x =_\tau y$$

So, whatever the domain of application, we may insist on a fixed relationship between $=_\tau$, $+_\tau$ and $-_\tau$.

The rules of inference of this logic are the natural deduction rules provided by the Isabelle theorem prover for HOL and are as documented in Nipkow et al. (2002)[ch5]. The semantics of the logic are described in Gordon and Pitts (1994). This semantics defines the concept of *interpretation* of an ontology O as giving a meaning to each $c{:}\tau \in Sig(O)$ and, hence, a truth value to each sentence $\phi \in Lang(O)$. We write $\mathcal{M} \models \phi$ if ϕ is assigned $\top$ by the interpretation $\mathcal{M}$. If $\forall \phi \in Ax(O).\ \mathcal{M} \models \phi$ then $\mathcal{M}$ is said to be a *model* of O. We can regard the real world as defining an interpretation that is intended to be a model of all our ontologies. We will refer to this interpretation as the *standard model*.

Definition 7 (Combining Ontologies) *The* combination*, $O_x \oplus O_y$, of two ontologies, O_x and O_y is defined as:*

$$O_x \oplus O_y \quad ::= \quad \langle Sig(O_x) \cup Sig(O_y), Ax(O_x) \cup Ax(O_y) \rangle$$

provided no clash of constant types arises, i.e.,

$$\forall c \in C.\ c{:}\tau_x \in Sig(O_x) \wedge c{:}\tau_y \in Sig(O_y) \implies \tau_x = \tau_y$$

$\oplus : \mathbb{O} \times \mathbb{O} \Rightarrow \mathbb{O}$ is a meta-constant.

Note that the language of the combined ontologies contains the union of their languages and the theorems of the combined ontologies contains the union of their theorems. If there is a clash of constants which have different meanings in the two ontologies combined, then the constants are renamed before combining in avoid the clash.

The associativity and commutativity of $\oplus$ follows trivially from the associativity and commutativity of $\cup$. We can, therefore, drop the parentheses from $O_1 \oplus (O_2 \ldots (O_{n-1} \oplus O_n) \ldots)$ and write $O_1 \oplus \ldots \oplus O_n$ unambiguously.

Definition 8 (Local *vs* Global (In)consistency) *Consider the set of ontologies $\{O_i : \mathbb{O} | 1 \leq i \leq n\}$. The set is said to be* locally (in)consistent *if each O_i is (in)consistent. It is said to be* globally (in)consistent *if $O_1 \oplus \ldots \oplus O_n$ is (in)consistent.*

Definition 9 (Ontology Fault) *By an* ontology fault *we mean one of the following two cases:*

Over-specification: It is possible to prove a false theorem*, i.e., $O \vdash \phi$ but $\mathcal{M} \models \neg\phi$, where $\mathcal{M}$ is a standard model of ontology O and ϕ is a sentence in $Lang(O)$.*

A special case of over-specification is inconsistency*, i.e., $O \vdash \bot$, where $\bot$ represents false, since $\top$ is true in all models. Note that inconsistency would arise if we tried to combine two ontologies, say O_x and O_y, where there exists a sentence ϕ such that $O_x \vdash \phi$ and $O_y \vdash \neg\phi$, since $\bot$ would then be a theorem of the combined ontology $O_x \oplus O_y$.*

Under-specification: It is impossible to prove a true sentence*, i.e., $O \nvdash \phi$ but $\mathcal{M} \models \phi$, where $\mathcal{M}$ is a standard model of ontology O and ϕ is a sentence in $Lang(O)$.*

A special case of under-specification is redundancy*, i.e., that $O \nvdash t_1 = t_2$ but $\mathcal{M} \models t_1 = t_2$, where $\mathcal{M}$ is a standard model of ontology O and t_1 and t_2 are terms of the same type in $Lang(O)$.*

3.3 Higher-Order Unification and Matching

Higher-order unification is the problem of unifying simply-typed λ-terms (Church, 1940), that $u\sigma$ and $v\sigma$ are equivalent w.r.t. α-, β- and possibly η-equivalences. Two expressions are α-equivalent if and only if the only difference between them is the renaming of bound variables, which essentially captures the principle that only the binding structure they induce matters; for instance, $\lambda x.\lambda y.2 \times y$ and $\lambda y.\lambda y.2 \times y$ are α-equivalent. Two expressions are β-equivalent if and only if they reduce to the same normal form, i.e. after substituting a function's arguments for its parameters; for instance, $(\lambda x.2 \times x)1$ is β-equivalent to 2×1. Lastly, two expressions are η-equivalent if and only if they are equivalent after adding or dropping abstractions; for instance, $\lambda x.f(x)$ is η-equivalent to f, if x does not appear free in f.

The method of solving equations by unification lays the foundations for numerous reasoning techniques and applications, including generalisations of resolution to second-order logic (Pietrzykowski, 1973), HOL programming languages such as λ-Prolog (Miller and Nadathur, 1986; Nadathur and Miller, 1998) and robust implementations of inference rules in theorem provers (Paulson, 1986).

Suppose $?f$ a second-order function, e.g., $?f(\lambda a.a)$, which takes a function as an argument. Unifying it with C may give infinitely many unifiers (Zaionc, 1985), including

$$\{?f \mapsto \lambda a.a(C),\ ?x \mapsto \lambda a.a\} \tag{3.2}$$

$$\{?f \mapsto \lambda a.a(a(C)),\ ?x \mapsto \lambda a.a\} \tag{3.3}$$

$$\{?f \mapsto \lambda a.a(a(a(C))),\ ?x \mapsto \lambda a.a\} \tag{3.4}$$

$$\ldots$$

Because of this potential issue, most higher-order reasoners and theorem provers fix a unification search bound, which prevents unification from searching past a certain depth.

For the unification of polymorphic simply-typed λ-terms, terms may contain type variables that need to be instantiated during the unification process. Suppose $?f$, $?x$ and C are of types $\alpha \Rightarrow \tau$, α and τ, respectively, where α is a type variable and τ is some base type. Any instantiation of α must be of the form

$$\tau_1 \Rightarrow \ldots \Rightarrow \tau. \tag{3.5}$$

which gives a potentially infinite set of solutions. Different instantiations of α may give rise to different instantiations of $?f$ and x. For instance, if α is instantiated to

$\tau \Rightarrow \tau$, then

$$?f \mapsto \lambda a.a(Y_1(a), Y_2(a)),\ ?x \mapsto \lambda a_1, a_2.C \tag{3.6}$$

is a solution, where Y_1 and Y_2 are new function variables, but it is not a solution if α is instead instantiated to $\tau \Rightarrow \tau \Rightarrow \tau$.

3.3.1 Huet's Algorithm

Whilst higher-order unification is undecidable in general, Huet gave a semi-decidable algorithm for pre-unification that, in effect, restricts the search space and finds a solution if it exists, but postpones certain solvable equations instead of enumerating their solutions (Huet, 1975, 2002). It is widely used and most implementations of higher-order unification adopt Huet's two procedures: SIMPL and MATCH, used for dealing with so called *rigid-rigid* and *flex-rigid* pairs of expressions, respectively. An expression is called rigid if its head[5] in β-normal form, which is if no β-reduction is possible, is a constant, free variable, or a bound variable; for instance, the expression $\lambda x_1, \ldots, x_n.\, F(s_1, \ldots, s_p)$ is rigid, provided that F is a constant, free variable or a bound variable. Otherwise, it's called flex, i.e. if F is a meta-variable (or logic-variable).

Recall that a unification problem formulated in the form $u =^? v$, which is typically expressed as a pair $\langle u, v\rangle$. A rigid-rigid pair is an equation where both u and v are rigid, e.g.,

$$\langle G(s_1, \ldots, s_p), H(t_1, \ldots, t_p)\rangle \tag{3.7}$$

where neither G nor H is a meta-variable. To unify a rigid-rigid pair, SIMPL essentially performs first-order unification. If the heads of the expressions are the same, i.e. $G = H$, then the input pair is simplified to a set of pairs $\{\langle s_1, t_1\rangle, \ldots, \langle s_p, t_p\rangle\}$. If the expressions have different heads, i.e. $G \neq H$, then SIMPL returns a failure status because the input pair is non-unifiable. The SIMPL step is repeated for all rigid-rigid pairs in the current problem, leaving only flex-rigid and flex-flex pairs for further unification.

The MATCH step is applied to the current problem if it contains at least one flex-rigid pair, e.g.,

$$\langle ?f(s_1, \ldots, s_p), F(t_1, \ldots, t_q)\rangle \tag{3.8}$$

where $?f$ is a meta-variable, and generates substitutions based on the *imitation* and *projection* rules. Imitation applies only if the head of the rigid term is a constant by generating a substitution that replaces the head of the flex term by a term equivalent to

[5] Given $\lambda x_1 \ldots . \lambda x_n.t\ M_1 \ldots M_m$, t is the head of the expression.

the head of the rigid term. Suppose F is a constant in (3.8), the imitation step generates a substitution of the following shape:

$$?f \mapsto \lambda x_1,\ldots,x_p.\, F(h_1(x_1,\ldots,x_p),\ldots,h_q(x_1,\ldots,x_p)) \tag{3.9}$$

where h_i for $1 \leq i \leq k$ are fresh meta-variables. In effect, the head of the instantiation imitates that of the rigid term. Projection applies if the head of the rigid term is either a constant or a bound variable by generating a substitution that has a term of the same target type as the flex term as its head that gives the flex term the correct type. Considering (3.8), the substitution generated by projection is given by:

$$?f \mapsto \lambda x_1,\ldots,x_p.\, x_i(h_1(x_1,\ldots,x_p),\ldots,h_k(x_1,\ldots,x_p)) \tag{3.10}$$

where h_i for $1 \leq i \leq k$ are fresh meta-variables. In this case, if the type of $?f$ is $\alpha_1 \Rightarrow \cdots \Rightarrow \alpha_p \Rightarrow \beta$, then the type of x_i, α_i, must be $\gamma_1 \Rightarrow \cdots \Rightarrow \gamma_k \Rightarrow \beta$.

When only flex-flex pairs remain, Huet's algorithm always reports a success status and maintains these pairs as constraints for later resolution.

The synchronisation of calls to SIMPL and MATCH is handled by the main procedure of the algorithm and a matching tree is constructed after repeated calls to these steps. Upon each call to MATCH, two nodes are generated – one for each imitation and projection. For each newly generated unification problem, the process is repeated.

3.3.2 Higher-Order Matching

Higher-order matching is a well-behaved fragment of higher-order unification. It is the problem of, given an equation $u =^? v$, pattern matching u to v, where v is closed, i.e. v does not contain meta-variables. Note that this is more restrictive than unifying terms in a flex-rigid pair, because only the head of the rigid expression cannot be a meta-variable in a flex-rigid pair. However, the decidability of higher-order matching is still an open problem. First-order matching is decidable and there is either one or zero solution to each matching problem. Second-order matching is also decidable (Huet and Lang, 1978) and the number of unifiers is finite[6]. At order three, it is, again, decidable (Dowek, 1994), but there may be an infinite number of unifiers (3.2, 3.3, 3.4). Fourth-order matching has also been shown to be decidable (Padovani, 2000), but there are no accurate results for greater orders.

[6]The number of unifiers is finite up to α-equivalence.

3.4 Isabelle

Isabelle is a generic interactive theorem prover (Nipkow et al., 2002) which encodes a range of object-logics in its intuitionistic, higher-order meta-logic, Pure. Some of the supported object-logics include Zermelo-Fraenkel set theory (ZF), first-order logic (FOL) and higher-order logic (HOL). Isar, which is Isabelle's declarative proof language, is presented to end users as a language for writing human-readable proof scripts. It is designed to fill the gap between the extremes of working with proof objects and natural language.

Isabelle follows the LCF approach (Gordon et al., 1979), so it has a very small core of code containing a small set of primitive inference rules, which guarantee correctness. As such, new theorems can only be derived from previously proven statements through the application of this set of simple inference rules. Isabelle also provides a range of powerful mechanisms for automating proofs, including the *auto*-tactic, simplification, and various kinds of resolution, including Sledgehammer, which interfaces Isabelle/HOL with automatic first-order provers such as E (Schulz, 2002), SPASS (Weidenbach et al., 2009), and Vampire (Riazanov and Voronkov, 2002). Isabelle supports polymorphic higher-order logic, augmented with axiomatic type classes. More precisely, though, Isabelle's λ-calculus is actually simply-typed, because $\Rightarrow$ is the only type constructor and it builds types from base types. However, it also resembles to some extent Hindley-Milner polymorphism (Hindley, 1969; Milner, 1978), the type system underlying ML and Haskell, because it, e.g., supports type-variables, although it has no let construct for defining polymorphic constants within terms.

A notion of modularity within Isabelle's logics is provided by axiomatic type classes (Wenzel, 1997), which allow types to be defined in terms of properties satisfying certain axioms. Terms and concepts can be introduced within an axiomatic type class, whilst theorems can be derived to hold for any type in the class. Another infrastructure for modular proof development is Isabelle's *locales*, which are designed for contextual reasoning. Locales are defined in terms of parameters that are fixed over a collection of assumptions. Parameters correspond to abstract constants, whereas assumptions correspond to axioms local to the locale.Thus, formal reasoning in a modular fashion can be done in Isabelle using locales (Ballarin, 2006). Locales act as independent proof contexts, yet they can be combined: the notion of extension can be realised by instantiating a locale inside another locale and giving values to the parameter variables; and, the notion of combination (Definition 7) by including multiple locales in a locale. Each

```
locale L =
  fixes te :: Object => Mom => real
  and ke :: Object => Mom => real
  and pe :: Object => Mom => real
  assumes ax: te(o, t) = ke(o, t) + pe(o, t)

locale L1 =
  L te1 ke1 pe1

locale L2 =
  L te2 ke2 pe2
```

Figure 3.1: Example locales specifying a definition of total energy, where ‘L’, ‘L1’ and ‘L3’ are the labels of three locales; ‘te’, ‘ke’ and ‘pe’ are the parameter variables of ‘L’; and, ‘ax’ is an axiom; both ‘L1’ and ‘L2’ depend on ‘L’, each with its own instantiation of ‘L’, and thus, the axiom ‘ax’.

locale has its own fixed set of parameters and axioms; an example locale specifying the equation of total energy is shown in Figure 3.1.

Multiple locales can be specified within a global context, so each locale can be viewed as a local, non-overlapping context. Essentially, a locale is implemented as a predicate and parameters as arguments of the predicate. The type of a locale with n parameters is the same as that of a n-ary predicate. For instance, a locale with three parameters of type α is $\alpha \Rightarrow \alpha \Rightarrow \alpha \Rightarrow bool$.

3.5 Summary

In this Chapter, we have presented some useful background knowledge for understanding the work in this thesis, including several theoretical components that are fundamental to the research into the design and implementation of GALILEO and a description of Isabelle. The formalisation of ontologies in HOL has provided both the syntax and semantics of the language in consideration. It has, in effect, highlighted the formal elements that potentially render our ontologies more flexible and expressive than those based on DL. However, the logic of higher-order functions is undoubtedly difficult. Although higher-order unification is undecidable, Huet’s algorithm generates a finite set of terms in second-order matching and decidability has been proven up to fourth-order. Nonetheless, the undecidability of higher-order unification is only a theoretical result and is not detrimental in most practical cases faced by GALILEO, because our use of Isabelle tends to present simplified problems to the unification algorithm.

Chapter 4

Ontology Repair Plans

4.1 Introduction

The process of revising an ontology in the face of new information is key to many areas of Computer Science. The literature on the subject calls such processes ontology evolution and we decided to investigate ontology evolution by formalising and mechanising a method to repair ontologies containing faults. For instance, we aim to repair locally consistent but globally inconsistent ontologies, i.e. ontologies that are individually consistent but may give rise to an inconsistency when merged (Definition 9, p.34). Working with locally consistent ontologies enables reasoning about the shape of the cause of the global conflict, allowing for specific meaningful repairs. The term *higher-order logic* is commonly used to refer to a logic in a type theory that is not dependent or polymorphic. However, polymorphism is particularly important to our work, so we base the underlying logic on an implementation of higher-order logic that supports polymorphism (§3.2.2, p.32). Unlike the less expressive logics, including DL, our approach naturally allows for the formalisation of ontology evolution both as belief revision and as syntactic manipulation, e.g., splitting/combining a function, changing its arity, etc. HOL has proven advantageous in at least three other ways. Firstly, Isabelle/HOL's polymorphism of variables and overloading of symbols such as $\leq$, $\geq$, $+$, $-$, etc. permits the generality of the described repair plans for evolution and their applicability over diverse cases. Secondly, HOL-based theorem provers, such as Isabelle, enable HO-reasoning for ontology evolution and reasoning across multiple ontologies, even over locally consistent but globally inconsistent ontologies that share symbols. Finally, many complex concepts are better represented as HOL objects, e.g., the orbit of a star – this is relevant because the examples used for developing and testing GALILEO are based on the evolution of Physics, which involves concepts best represented as functions. This, therefore, relates our work to scientific discovery (§2.5, p.24), e.g., that reported by Langley (1981).

We have developed a series of *ontology repair plans* (ORP) which operate on a small set of modular higher-order ontologies, e.g., one representing an initial theory of Physics and another representing a particular experimental set-up. Each repair plan has a trigger formula and some actions: when the trigger is matched, the actions are performed. The mechanisms we have developed are called *plans*, because each ORP explicitly specifies the pre-conditions for a repair and the post-condition for a repair is to have the detected conflict eliminated, i.e. the corresponding trigger formula becomes no longer provable. This is, therefore, similar to STRIPS-style plan operators (Fikes and Nilsson, 1972). The actions modify both the signatures and the axioms of the old ontologies to produce new ones. A principle of the design and the formalisation of the ORPs is to ensure sufficient generality, yet succinctness, in the descriptions. The rest of this chapter describes each of these ORPs in depth by describing both the trigger formulae and repair rules.

In the rest of this chapter, we will also present the examples used to motivate the structure of each ORP – each of which has been treated as a *development* case study.

4.2 Representing Ontology Repair Plans

The general notion of ORP in terms of the logic we have adopted is the following.

Definition 10 (Ontology Repair Plans) *An* ontology repair plan *is a pair*

$$\langle Trigger, Repair \rangle \tag{4.1}$$

where:

$Trigger$**:** *is a set of meta assertions of the form* $O \vdash \phi$ *or* $O \nvdash \phi$ *which collectively describe an ontology fault.*

$Repair$**:** *is a set of repair operations of the form* $\nu(O) ::= \pi(O)$*, where* $\nu(O)$ *denotes a repaired ontology and* π *is an operation on the signature and/or axioms of the* *original ontology O. If each O is replaced by $\nu(O)$ in $Trigger$ to form $\nu(Trigger)$ then $\nu(Trigger)$ no longer describes a fault that the ontologies suffer from* [1].

Unprovability is an undecidable problem, and our approach to showing $O \nvdash \phi$ is discussed in §4.6.

Note that the signature of the repaired logic, $Sig(\nu(O))$, may contain type declarations for constants that are not declared in the signature of the original ontology, $Sig(O)$, and vice versa. One consequence is that a model of O may not even be an interpretation of $\nu(O)$ and vice versa. This makes it difficult to give a model-based semantics to the operations of ontology repair plans. Note that this problem is caused by the inclusion of signature change in ontology evolution and not by the use of higher-order logic. We are exploring other possibilities, but assigning a semantics to our version of higher-order ontology evolution remains further work.

In the following sections we present the formalisations of five different ORPs, each is designed to repair a unique kind of ontological fault: *Where's My Stuff*, *Reidealisation*, *Inconstancy*, *Unite* and *Spectrum*.

4.3 The *Where's My Stuff* Ontology Repair Plan

In the basic ontological setup, which consists of a single predictive theory and a single set of empirical observations or evidence, a common type of conflict in Physics is caused by a difference between the predicted value of some property and the value according to some corresponding sensory information arising from an experiment. We believe that this type of conflict indeed regularly occurs throughout the evolution of Physics as we have identified the most number of historical records (15, so far) associated with it. This conflict is typically caused by the use of a theoretical definition that is based on an incorrect definition of the property measured or on an incorrect definition of a dependency of the property measured. The error is generally a consequence of a misconceptualisation that neglects a component of the property from the definition; for instance, when speaking about the total amount of heat in a body, the notion of latent heat is missing from the theoretical definition of the total heat.

[1] Although, other ontology faults may still remain to be detected and repaired.

4.3.1 Motivating Example: The Discovery of Latent Heat

Until the second half of the 18^{th} century, the chemical/physical notion of heat was conflated with the notion of temperature and it was seen as a function of time. Such a view of heat can be interpreted as understanding heat transfer as a flow. Flow was defined as occurring when two physical bodies at different temperatures were in direct contact with one another. Equation 4.2 is a rational reconstruction of this pre-modern view:

$$\Delta Temp = \Delta Q = k \times m \times \Delta t \tag{4.2}$$

where:

- $\Delta Temp$ is the change in temperature
- ΔQ is the total amount of energy released or absorbed
- k is an adjustment factor depending on the material
- m is the mass of the object
- Δt is the duration of the flow
- the polarity of $Temp$ is positive if the object is being heated and negative if it is being cooled.

In Equation 4.2, $\Delta Temp = \Delta Q$ means that the notions of temperature and heat were conflated. As to be discussed further in Chapter 5, a Physics equation is typically expressed as a mathematical equation, which asserts an equality between two expressions; these expressions may contain variables for values that could change in the given problem. In a logical formulation, dependent variables should not be represented by logical variables but should instead be represented as functions, as functions take some arguments and return a result that depends on these arguments. A formulation of 4.2 in HOL[2], with dependent variables replaced by functions and quantifying over the domain, is therefore:

$$\begin{aligned} &TempDiff(o,e) ::= \\ &\quad HeatDiff(o,e) = k \times Mass(o) \times Period(Start(e), End(e)) \end{aligned} \tag{4.3}$$

where:

[2]The representation can instead be formalised in a typed first-order logic in this example.

- $(TempDiff :: Obj \Rightarrow Event \Rightarrow \mathbb{R})(o, e)$ returns the amount of change in temperature in an object o as a result of undergoing event e;
- $(HeatDiff :: Obj \Rightarrow Event \Rightarrow \mathbb{R})(o, e)$ returns the amount of change in heat in an object o during an event e;
- $(Start :: Event \Rightarrow Mom)(e)$ returns the moment at the start of an event e;
- $(End :: Event \Rightarrow Mom)(e)$ returns the moment at the end of an event e;
- $(Mass :: Obj \Rightarrow \mathbb{R})(o)$ returns the mass of an object o, assuming the mass of o is constant for the sake of simplicity;
- $(Period :: Mom \Rightarrow Mom \Rightarrow Duration)(m_1, m_2)$ returns the duration between moments m_1 and m_2;
- *Event* is the type of event;
- *Mom* is the type of time moment; and,
- $k : \mathbb{R}$ is a real constant used as an adjustment value.

In other words, the change in temperature (likewise, in heat) was regarded to be directly proportional to the length of time the object was exposed to the event, which means the longer an object is heated, the hotter it gets. The function *TempDiff* takes an event as an argument because its value depends on the length of the event under consideration. Similarly, for *HeatDiff*.

Joseph Black discovered the concept of latent heat around 1750. Wiser and Carey (1983) discuss a period when heat and temperature were conflated, which presented a conceptual barrier that Black had to overcome before he could formulate the concept of latent heat. This conflation creates a paradox: as ice is melted it is predicted to gain heat, but its heat, as measured by temperature, remains constant. Black had to split the concept of heat into two separate concepts: latent heat, which is the heat exchanged during phase change, and sensible heat, which is responsible for a change in temperature.

Suppose there are two ontologies: *Pred*, the ontology containing the predictive theory, and *Obs*, the ontology containing observations. The axiomatisation of the case study can be formalised as Figure 4.1.

$$Ax(Pred) \supseteq \{$$

$$\forall o{:}Obj, e{:}Event.HeatDiff(o,e) ::= \tag{4.4}$$

$$k \times Mass(o) \times Period(Start(e), End(e)),$$

$$Mass(H_2O) > 0, \tag{4.5}$$

$$Period(Start(Melting), End(Melting)) = 10, \tag{4.6}$$

$$Process(H_2O) = Melting, \ \ldots \tag{4.7}$$

$$\}$$

$$Ax(Obs) \supseteq \{$$

$$HeatDiff(H_2O, Process(H_2O)) = 0 \tag{4.8}$$

$$\}$$

where:

- *Pred* is the predictive ontology containing the theory
- *Obs* is the empirical ontology containing observations
- H_2O is the water (ice) being melted
- *Melting* is the melting event
- $Process(o)$ returns the process o undergoes

Figure 4.1: Axiomatisation of a representation of the discovery of latent heat.

In (4.7), $Process(H_2O)$ returns a melting event, because the water under consideration undergoes melting. The term $HeatDiff(H_2O, Process(H_2O))$, therefore, returns the amount of change in the heat in H_2O during the melting event.

The paradox faced by Black can be inferred from the two ontologies as follows:

$$Pred \vdash HeatDiff(H_2O, Process(H_2O)) > 0 \tag{4.9}$$

$$Obs \vdash HeatDiff(H_2O, Process(H_2O)) = 0 \tag{4.10}$$

Equation 4.9 is deduced from the predictive, physical theory that heat increases strictly monotonically when objects are warmed, along with some basic arithmetic, and 4.10 comes from the observed constant temperature during the melting, which is specified as an assertion in this particular example.

To resolve the conflict, one fix is to introduce the concept of latent heat of fusion in *Pred* and define it to be the difference between the total amount of heat and the amount of sensible heat. The new definition could be:

$$\forall o{:}Obj, e{:}Event.\ LHFDiff(o,e) ::= HeatDiff(o,e) - SHDiff(o,e) \tag{4.11}$$

in anticipation of their intended meanings, where *SHDiff* and *LHFDiff* can be read as the change in sensible heat and the change in the latent heat of fusion, respectively. The new definition means that the amount of change in latent heat is defined as the difference between the change in the total amount of heat in the object and the change in the amount of heat for raising the temperature. Equation (4.11) gives the same shape as the modern formulation of latent heat:

$$m \times L = \Delta Q - c \times m \times \Delta T \tag{4.12}$$

where

- latent heat is defined as $m \times L$
- sensible heat is defined as $c \times m \times \Delta T$

and

- m is the mass of the object[3]
- L is the specific latent heat for the substance of the object
- ΔQ is the total amount of energy released or absorbed
- c is the specific heat capacity for the substance of the object
- ΔT is the change in temperature of the object.

Beside the new definition, occurrences of *HeatDiff* in *Obs* should be renamed to *SHDiff* to indicate that the kind of heat observed is only a part of the total amount of heat.

The definition introduced by the repair (4.11) is not precisely what is required, but is along the right lines. Some further indirect observations of *LHFDiff* are required to witness its behaviour under different states of o so that it can be further repaired, e.g., the removal of its e argument. The *SHDiff* part of the new definition needs to be further refined so that its contribution of heat depends both on temperature and mass instead.

[3] This is not the same as the m used in the logical formulation.

4.3.1.1 Discussion

From this case study, we can see that a number of features must be incorporated into WMS ,and also into other ORPs, in order to increase their generalities. For instance, the term that is deemed responsible for the fault, which we call *stuff*, should be variadic, as we would want it to be instantiated to, e.g., *Heat*, which is a ternary function, or *Thermometer*, which is a binary function. We also want *stuff* to be able to be instantiated to some arguments of the dominant function, e.g., instantiate *stuff* to H_2O when given the term $Heat(H_2O, Process(H_2O))$, so the variable *stuff* should be positioned as an argument of some function in the trigger formula rather than being a dominant function itself. Further, *Heat* and *Process* have different target types, e.g., one returns a real while the other returns an event, so the equal operator (and other relevant operators) should be polymorphic.

4.3.2 Overview of *Where's My Stuff*

Useful features of the conflict and repair underlying the discovery of latent heat can be captured and generalised to create a general ORP, which we call *Where's My Stuff*. The *Where's My Stuff* ORP (WMS), formally represented in Figure 4.2, assumes that we have two ontologies and is triggered when the return value of a function deduced from one ontology conflicts with the return value of the same function deduced from another ontology. Suppose we have two ontologies, O_1 and O_2, in conflict with each other: O_1 may contain the current state of a predictive theory and O_2 may contain some observation or empirical evidence or vice versa. Suppose a measurement function f measures some property of *stuff* and WMS is triggered if the return value of $f(stuff)$ deduced from O_1 is different from that deduced from O_2 (4.13, 4.14). The two variables in $f(stuff)$, *stuff* and f, which themselves are polymorphic, respectively represent the part of the term that is responsible and not responsible for the conflict; therefore, formulae containing occurrences of the *stuff* require amendment. WMS resolves the detected conflict by splitting *stuff* into three parts: visible *stuff*, invisible *stuff*, and total *stuff*, and defining invisible *stuff* in terms of total and visible *stuff*s in the repaired O_1, $\nu(O_1)$ (4.15, 4.18). The new O_2, $\nu(O_2)$, is the same as O_2 except for the renaming of *stuff* to $stuff_{vis}$ (4.19).

Suppose two ontologies, $O_1{:}\mathbb{O}$ and $O_2{:}\mathbb{O}$, are elements of a working set of ontologies, W, disagree over the return value of $f(stuff)$ of type τ':

Trigger: If $f(stuff)$ has a larger value in O_1 than in O_2 then the following formula will be triggered:

$$\exists O_1, O_2 \in W,\ \tau, \tau'{:}Types,\ f{:}\tau \Rightarrow \tau',\ stuff{:}\tau,\ v{:}\tau'.$$

$$(O_1 \vdash f(stuff) > v \wedge O_2 \vdash f(stuff) \leq v) \vee \tag{4.13}$$

$$(O_1 \vdash f(stuff) \geq v \wedge O_2 \vdash f(stuff) < v) \tag{4.14}$$

where O_1 and O_2 share a set of base types; $O \vdash \phi$ means that formula ϕ is a theorem of ontology O; $t{:}Types$ means t is a type; $o{:}Onto$ means o is an ontology; $>$ is a partial order for τ'. In the case where each partial order in (4.13, 4.14) is reversed, the roles of O_1 and O_2 below are reversed.

Repair: We introduce two new kinds of stuff, *visible* and *invisible* stuff, and create a definition of *invisible* stuff.

$$stuff_{invis} := stuff - stuff_{vis} \tag{4.15}$$

Let $\nu(O_1)$ and $\nu(O_2)$ be the repaired ontologies. The signatures of the new ontologies are defined in terms of those of the old as follows:

$$Sig(\nu(O_1)) \quad := \quad \{\ stuff_{vis}{:}\tau, stuff_{invis}{:}\tau\ \} \cup Sig(O_1) \tag{4.16}$$

$$Sig(\nu(O_2)) \quad := \quad \{\ stuff_{vis}{:}\tau\ \} \cup Sig(O_2) \tag{4.17}$$

We revise the axioms for the new ontologies in terms of those of the old as follows:

$$Ax(\nu(O_1)) \quad := \quad \{\ stuff_{invis} = stuff - stuff_{vis}\ \} \cup Ax(O_1) \tag{4.18}$$

$$Ax(\nu(O_2)) \quad := \quad \{\ \phi\{stuff_{vis}/stuff\} \mid \phi \in Ax(O_2)\ \} \tag{4.19}$$

where $Ax(O)$ is the set of axioms of ontology O. To effect the repair, the axioms of $\nu(O_1)$ are the same as those of O_1 except for the addition of the new definition; the axioms of $\nu(O_2)$ are the same as those of O_2 except for the renaming of the original stuff to the visible stuff.

Figure 4.2: The "Where's My Stuff?" ontology repair plan

4.3.3 Discussion

In order for WMS to have a sufficiently high generality, f and *stuff* are of polymorphic types, so that *stuff* can be instantiated to individuals, functions, or predicates; for instance, a particular star (individual), the orbit of a star (function), or a galaxy (predicate or set). As such, the partial orders for the return values of $f(stuff)$, e.g., $<$, $\leq$, $>$, and $\geq$, also need to be polymorphic. For example, if τ' is instantiated to $\mathbb{R}$, the type for real numbers, then $\leq_{\mathbb{R}}$ is a partial order on reals, whereas if τ' is instantiated to $\mathbb{R} \Rightarrow \mathbb{R}$, the type for functions taking a real and returning a real, then $\leq_{\mathbb{R}\Rightarrow\mathbb{R}}$ is a partial order on functions taking a real and returning a real.

There are two (subtly) different ways to identify a WMS-type of conflict: where $f(stuff)$ is strictly greater than some value in O_1 but is equal or less than it in O_2 (4.13), and where $f(stuff)$ is greater than or equal to some value in O_1 but is strictly less than a particular value in O_2 (4.14). In our earlier reports (Bundy and Chan (2008)), we adopted a trigger formula in the following style:

$$\exists O_1, O_2{:}\mathbb{O},\ \tau, \tau'{:}Types,\ f{:}\tau \Rightarrow \tau',\ stuff{:}\tau,\ v_1, v_2{:}\tau'. \qquad (4.20)$$
$$O_1 \vdash f(stuff) = v_1\ \wedge\ O_2 \vdash f(stuff) = v_2\ \wedge O_1 \vdash v_1 > v_2$$

along with the same repair rules as (4.16 - 4.19), and the roles of O_1 and O_2 are also reversed for repair if $O_1 \vdash v_1 < v_2$. It might appear that (4.20) already suffices to detect a conflict between O_1 and O_2 as (4.20) can be matched if $f(stuff)$ returns a value higher in O_1 than that in O_2. However, the coverage of this trigger formula is not sufficiently comprehensive, because the exact value of $f(stuff)$ is not guaranteed to be deducible from either ontology. For instance, if we are given the following axiomatisation:

$$Ax(M) \ := \ \{\ f(stuff) > 0\ \} \qquad (4.21)$$
$$Ax(N) \ := \ \{\ f(stuff) < 0\ \} \qquad (4.22)$$

then, clearly, this is a conflict that is compatible with that intended to be addressed by WMS, as the return value of $f(stuff)$ in M, which is strictly greater than 0, must be greater than that of $f(stuff)$, which is strictly less than 0, in N; however, the exact value of $f(stuff)$ on either side is not deducible.

4.4 The *Reidealisation* Ontology Repair Plan

Given two ontologies that disagree over the return values of some measurement function, which is the kind of conflict WMS detects and repairs, one could resolve a special case of this kind of conflict without inventing some invisible component, $stuff_{invis}$, and taking the viewpoint that the original conceptualisation gives only a partial view of the underlying property. Instead, the *idealisation* of the property could be changed such that it is viewed as being a property of another type. With the new idealisation, the measurement function, which originally returns different values in each of the two ontologies, is expected to no longer return conflicting values. We do not want to break the contradiction by arbitrarily changing the idealisation, e.g., assigning the property to a type that is not in the type hierarchy of the domain of the measurement function. The new type should, therefore, be one that is in the type hierarchy of the domain of the measurement function and does not result in contradicting measurement values. To determine the new type assignment to the property, one could project a new type assignment and verify that it would not result in a contradiction based on the existing Physics. For instance, if some measurement function gives contradicting values for a bouncing ball when it is idealised as a particle without extent, then the idealisation could be changed and checked against the existing Physics to ensure consistent measurement values are returned if it is regarded to be a spring.

4.4.1 Motivating Example: Bouncing-ball Paradox

The *bouncing-ball paradox*, which is described by diSessa (1983), considers the situation in which a ball is dropped from above ground and a student is asked to predict the amount of its total energy when it makes impact with the ground. Suppose this ball is a "perfect" ball, where there is no loss of energy in each bounce. The student takes a (wrong) definition of total energy and defines it to be the sum of kinetic energy and potential energy. The kinetic energy is initially zero, as it is held stationary, and the potential energy is initially greater than zero, as the ball is held above ground. By the law of conservation of energy, the total amount of an energy in an isolated system is constant, so the final total energy of the system containing the ball can be calculated. Since we assume the system to contain just the ball, the final total energy of the ball can be inferred. The paradox is exactly the discrepancy between the initial and final amounts of total energy of the ball, as the ball is elastic but the definition applied to calculate the total energy is the one for particles *without* extent.

Suppose we have two ontologies *Student*, containing the student's theory about the Physics, and *Obs*, containing the experimental results about the ball *Ball* in a thought experiment and the system containing it. Let *TE*, *KE*, *PE* and *EE* denote polymorphic functions for calculating the total energy, kinetic energy, potential energy and elastic energy[4] of particles and springs (with extent), respectively. Suppose we can derive the following in the two ontologies

$$Student \vdash TE(x,e) = \begin{cases} KE(x,e)+PE(x,e)+EE(x,e) & \text{if } x{:}Spring \\ KE(x,e)+PE(x,e) & \text{if } x{:}Particle \end{cases} \tag{4.23}$$

$$\vdash TE(Ball{:}Particle, End(Drop)) > 0 \tag{4.24}$$

$$Obs \vdash TE(x,e) = \begin{cases} KE(x,e)+PE(x,e)+EE(x,e) & \text{if } x{:}Spring \\ KE(x,e)+PE(x,e) & \text{if } x{:}Particle \end{cases} \tag{4.25}$$

$$\vdash TE(Ball{:}Particle, End(Drop)) = 0 \tag{4.26}$$

$$\nu(Student) \vdash TE(Ball{:}Spring, End(Drop)) > 0 \tag{4.27}$$

$$\nu(Obs) \vdash TE(Ball{:}Spring, End(Drop)) > 0 \tag{4.28}$$

where *Particle* and *Spring* are, respectively, the type of particles and the type of springs; (4.23) and (4.25) are the definition of the total energy of particles and springs which, for springs, is defined as the summation of all kinetic, potential, and elastic energies, whereas, for particles, is defined as the summation of only kinetic and potential energies; and, e and x are universally quantified variables. Note that there is significant duplication in the different ontologies but they can be concisely represented in Isabelle, as discussed in §5.5. When the ball is idealised as a particle, *Student* and *Obs* give conflicting values for the amount of total energy. However, if the ball is idealised as a spring, the correct definition of total energy gives consistent values across *Student* and *Obs*. Notice that 4.27 and 4.28 represent post-conditions of the repaired ontologies.

Since we know from (4.27) and (4.28) that a contradiction can be avoided if the *Ball* is idealised as a spring, then it is sufficient to overwrite it with a new symbol representing the ball but of type *Spring* rather than *Particle*. A bouncing-ball could be idealised as a spring, as both are elastic objects, i.e. that they both can be deformed. This, therefore, results in the following repair to the signatures:

$$Sig(\nu(Student)) ::= \{\, \nu(Ball){:}Spring \,\} \cup Sig(Student) \setminus \{Ball\} \tag{4.29}$$

$$Sig(\nu(Obs)) ::= \{\, \nu(Ball){:}Spring \,\} \cup Sig(Obs) \setminus \{Ball\} \tag{4.30}$$

[4] The amount of elastic energy in particles (without extent) is zero.

where $\nu(Student)$ and $\nu(Obs)$ are the repaired *Student* and *Obs* ontologies, respectively; $Sig(\nu(Student))$ and $Sig(\nu(Obs))$ denote the sets of signature declarations in $\nu(Student)$ and $\nu(Obs)$. As with the axioms, occurrences of *Ball* should be replaced by $\nu(Ball)$.

4.4.2 Overview of Reidealisation

The idea of changing the idealisation of some property in order to avoid a contradiction, which would otherwise arise, can be applied to the design of another ORP, which we call *Reidealisation*. Reidealisation assumes that we have two ontologies and is triggered if two conditions are satisfied: i) the return values of a function f measuring some property *stuff* in each of these ontologies contradict each other, and ii) if the property of *stuff* is projected to be of some other type, then the same measurement function f does not return contradicting values. The trigger formulae are formalised in Figure 4.3, where the approach to detecting contradicting return values of $f(stuff)$ is similar to WMS. There are, altogether, two ways to check that $f(stuff)$ returns contradicting values between O_1 and O_2, as formalised in (4.31) and (4.32). To this end, we test the outcome of applying f to *stuff* when its type is assigned to τ_2. Since we define an ontology as a pair comprised of its set of signature elements and its set of axioms (Definition 6, p.32), to check whether two ontologies are the same is a matter of matching pairs of sets.

Suppose we have two ontologies O_1 and O_2, in which the values of $f(stuff)$ disagree if *stuff* is of type τ_1. Further in these ontologies, $f(\nu(stuff))$ return consistent values if $\nu(stuff)$ is of type τ_2. Since no contradiction arises if $\nu(stuff)$ is of type τ_2, the repair required is simply to overwrite *stuff* with a new term $\nu(stuff)$ and assign its type to τ_2 (4.34) in both O_1 and O_2. Occurrences of *stuff* are replaced by $\nu(stuff)$ in order to reflect the repair. We assume that if *stuff* appears as an argument of some function in an axiom, then the type of the corresponding argument position is a variable. So, the function itself must be polymorphic. The reason for this assumption is that *stuff* may already occur elsewhere within the axioms of an ontology, so replacing *stuff* with another term of a different type may lead to type errors. For instance, suppose $F{:}\mathbb{R} \Rightarrow bool$ and $c{:}\mathbb{R}$, $F(c)$ is a valid expression. However, if we let *stuff* be instantiated to c and force its type to change to, e.g., $\mathbb{N}$, then it would result in a type error unless the type of F is also changed.

Suppose two ontologies, O_1 and O_2, disagree over the return value of $f(\mathit{stuff})$ of type τ':

Trigger: If $f(\mathit{stuff})$, where f is overloaded, has two different values in O_1 and O_2 then the following formula will be triggered:

$$\forall\tau{:}C.\ \exists O_1, O_2{:}\mathbb{O},\ \tau_1 \subseteq \tau, \tau_2 \subseteq \tau,\ \tau'{:}Types,\ \mathit{stuff}{:}\tau_1,$$
$$f{:}\tau \Rightarrow \tau', v{:}\tau'.$$
$$(O_1 \vdash f(\mathit{stuff}) = v \land O_2 \vdash f(\mathit{stuff}) \neq v \lor \tag{4.31}$$
$$O_1 \vdash f(\mathit{stuff}) \neq v \land O_2 \vdash f(\mathit{stuff}) = v) \land \tag{4.32}$$
$$\nu(O_1) \vdash f(\nu(\mathit{stuff})) = v \land \nu(O_2) \vdash f(\nu(\mathit{stuff})) = v \tag{4.33}$$

where C is a type class in which f is a method; Formulae (4.31) and (4.32) require that when f is applied to *stuff* of type τ_1, a conflict arises, but when the same f in the repaired ontologies, $\nu(O_1)$ and $\nu(O_2)$, is applied to the repaired *stuff*, $\nu(\mathit{stuff})$, which is of type τ_2, the conflict disappears (4.33). One key feature of the trigger pattern of Reidealisation is that f must be overloaded, i.e. f must be defined over several different types, acting in a different way for each type.

Repair: Let $\nu(O_1)$ be the repaired O_1. We replace occurrences of $f(\mathit{stuff})$ with $f(\nu(\mathit{stuff}))$:

$$Sig(\nu(O_1)) := \{\ \nu(\mathit{stuff}){:}\tau_2\ \} \cup Sig(O_1) \setminus \{\mathit{stuff}\} \tag{4.34}$$
$$Sig(\nu(O_2)) := \{\ \nu(\mathit{stuff}){:}\tau_2\ \} \cup Sig(O_2) \setminus \{\mathit{stuff}\} \tag{4.35}$$
$$Ax(\nu(O_1)) := \{\ \phi\{\nu(\mathit{stuff})/\mathit{stuff}\ \} \mid \phi \in Ax(O_1)\ \} \tag{4.36}$$
$$Ax(\nu(O_2)) := \{\ \phi\{\nu(\mathit{stuff})/\mathit{stuff}\ \} \mid \phi \in Ax(O_2)\ \} \tag{4.37}$$

where $Sig(\nu(O))$ is the set of signature declarations of the repaired O and $Ax(\nu(O))$ is the set of axioms of the repaired O.

Figure 4.3: The Reidealisation ontology repair plan

4.4.3 Discussion

As depicted in Figure 4.3, f must be defined over at least types τ_1 and τ_2. Indeed, f is a polymorphic function, but there are two classes of polymorphisms: *parametric* and *ad-hoc* (3.2). The function f must, therefore, be ad-hoc polymorphic. Fortunately, Isabelle/HOL supports both parametric polymorphism and ad-hoc polymorphism, which is achieved through using type classes. Ad-hoc polymorphism is essentially achieved by splitting the introduction of the polymorphic function from its overloaded definitions, where each overloaded definition corresponds to the declaration of an instance of the class.

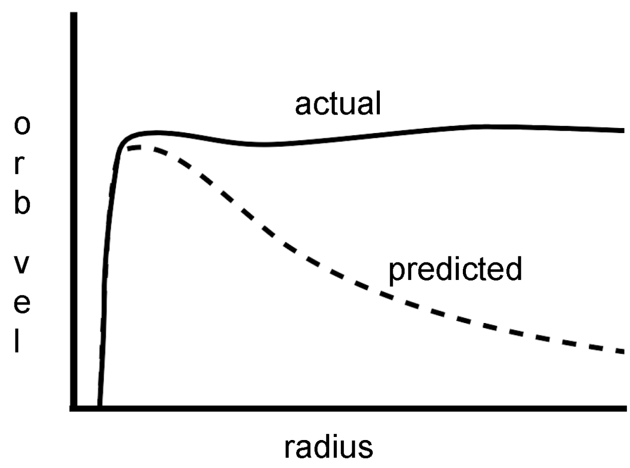
The repair rules (4.34 - 4.37) involve overwriting $stuff$:τ_1 with a new symbol $\nu(stuff)$:τ_2 and replacing occurrences of *stuff* with $\nu(stuff)$. We assume that the functions with *stuff* as an argument must be polymorphic. It may be a strong assumption in certain application, but we argue that it is reasonable in Physics. Physics properties tend to be generic concepts, applicable to a range of substances, materials, dimensions, and so forth. Nonetheless, the assumption could be relaxed by incorporating more complex transformations. Since the main objective of Reidealisation is to let $f(stuff{:}\tau_1)$ be evaluated as $f(stuff{:}\tau_2)$, we could alternatively introduce a new symbol $\nu(stuff)$:τ_2 *without* retracting $stuff$:τ_1 from the signature and replacing occurrences of *stuff* with $\nu(stuff)$ only within the axioms used in the inference. However, functions that take *stuff* as arguments and occur within those axioms would still need to be polymorphic. With *stuff* remaining in the signature, monomorphic functions that take *stuff* as arguments and are not part of the inference would still be well-formed. However, this requires some additional facilities in the reasoner in order to pinpoint those axioms.

Another approach is to generalise the type of all functions that take *stuff* as arguments to obtain a polymorphic input type; for instance, if $F{:}\mathbb{R} \Rightarrow bool$ and is applied to *stuff*, then the new type of F becomes $\alpha \Rightarrow bool$, where α is a type variable. However, this approach requires type changes to be propagated to symbols beside *stuff*. Providing sufficiently general, yet succinct, descriptions of the ORP is an important contribution of our work, so we have designed Reidealisation in a way without the need to specify the propagation of type changes. Nonetheless, the current formalisation of Reidealisation is adequate for emulating ontology evolution in Physics, but it can be naturally adapted if a more rigorous approach is preferred in the domain of application.

4.5 The *Inconstancy* Ontology Repair Plan

As previously described, WMS detects and repairs a commonly occurring type of logical conflicts that arises between a theory and some sensory information, which is the consequence of some sentence ϕ being derivable from one ontology and $\neg\phi$ being derivable from another. However, in order to better define a repair operation, extra sensory information can provide more accurate meanings to the repaired terms (Chan and Bundy, 2008). In practice, physicists often can obtain various experimental data gathered under different environments, so it is highly plausible that multiple sets of data can be used to challenge a Physics theory. We have identified from historical records of Physics development that the inclusion of additional sensory information collected under different conditions can indeed help identify new dependencies for a property.

4.5.1 Motivating Example: Modified Newtonian Mechanics



This diagram is taken from `http://en.wikipedia.org/wiki/Galaxy_rotation_problem`. *The x-axis is the radii of the stars and the y-axis is their orbital velocities. The dotted line represents the predicted graph and the solid line is the actual graph that is observed.*

Figure 4.4: Predicted *vs* Observed Stellar Orbital Velocities

Evidence for the existence of dark matter comes from various sources, for instance, from an anomaly in the orbital velocities of stars in spiral galaxies identified by Rubin et al. (1980). Given the observed distribution of mass in these galaxies, we can use Newtonian Mechanics to predict that the orbital velocity of each star should be inversely proportional to the square root of its distance from the galactic centre (called its *radius*). However, observations of these stars show their orbital velocities to be roughly constant and independent of their radius. Figure 4.4 illustrates the predicted and actual graphs. In order to account for this discrepancy, the MOdified Newtonian Dynamics (MOND), developed by Milgrom (1983), hypotheses that the Gravitational constant G is not a constant, but a function depending on the acceleration of the star. To identify a variation in G, we need to collect evidence for a set of stars, $\{star_i \mid 1 \leq i \leq n\}$ where n is the number of stars in the data sample . Suppose NMT and $Obs(Acc(star_i) = a_i)$ are the ontologies containing the predictive theory, i.e. Newtonian Mechanics, and the observations describing the situation of star $star_i$, whose acceleration, $Acc(star_i)$, is a_i, respectively. Suppose we work with only two stars, $star_1$ and $star_2$, we can formalise this example as:

$$NMT \vdash G = 6.673 \times 10^{-11} \tag{4.38}$$

$$Obs(Acc(star_1) = a_1) \vdash G = g_1 \tag{4.39}$$

$$Obs(Acc(star_2) = a_2) \vdash G = g_2 \tag{4.40}$$

$$NMT \vdash g_1 \neq g_2 \tag{4.41}$$

where $Acc(star_i)$ returns the acceleration of star $star_i$; $Obs(Acc(star_i) = a_i)$ is an ontology containing sensory information collected under the condition that the accelerations of $star_i$ is a_i; and, g_1 and g_2 are constants of different values. Note that the values of G in (4.39, 4.40) are assumed to be computable by some function. The fix proposed by MOND is to turn G from a constant into a function and add to it the acceleration of the star, $Acc(star_i)$, as an argument. The repaired ontologies would, therefore, be:

$$\nu(NMT) \vdash G = \lambda y.\ F(6.673 \times 10^{-11}, Acc(y)) \tag{4.42}$$

$$\nu(Obs(Acc(star_1) = a_1) \vdash G(Acc(star_1)) = g_1 \tag{4.43}$$

$$\nu(Obs(Acc(star_2) = a_2) \vdash G(Acc(star_2)) = g_2 \tag{4.44}$$

$$\nu(NMT) \vdash g_1 \neq g_2 \tag{4.45}$$

where F is a new function, whose value we can seek to determine by regressing against the data from the sensory ontology.

4.5.2 Overview of Inconstancy

As described in §4.5.1, the detection of an unexpected variation and the proposed fix to G require reasoning over sensory information collected under different conditions, i.e. over the values of G when $Acc(star_1) = a_1$ and when $Acc(star_2) = a_2$. To this end, it is essential to adopt a different modelling approach from that for WMS and Reidealisation, because the representation of the conditions underlying the data in the model is more demanding. Because we argue that theories for Physics and Mathematics can be more naturally engineered into a modular form, we now argue that data collected under different conditions should also be modelled by a modular approach. For example, if ψ holds under condition $\vec{cond}_1$ and $\neg\psi$ holds under condition $\vec{cond}_2$, then we have two ontologies, $O(\vec{cond}_1)$ and $O(\vec{cond}_2)$, which represent those containing the information collected under conditions $\vec{cond}_1$ and $\vec{cond}_2$, respectively, and that $O(\vec{cond}_1) \vdash \psi$ and $O(\vec{cond}_2) \vdash \neg\psi$. We assume that the condition vectors $\vec{cond}_1$ and $\vec{cond}_2$ specify only the conditions that have changed over the two snapshots.

The *Inconstancy* ORP, shown in Figure 4.5, assumes that there is one theoretical ontology and at least two sensory ontologies and is triggered when a function is predicted in the theory to be independent of some parameter, whereas a dependency on that parameter can be inferred from the two or more sensory ontologies. Suppose there is one ontology O_x, which may contain the predictive Physics theory, and at least two ontologies under distinct conditions, $O_{y,i}(b_i(v) = v_i) \oplus O_z$ and $O_{y,i}(b_j(v) = v_j) \oplus O_z$, which may contain observations made under conditions $b_i(v) = v_i$ and $b_j(v) = v_j$, where v is a function that measures some property of b_i and b_j. The $O_{y,i}(b_i(v) = v_i)$ component of the combination represents the maximal part of the sensory ontology that is not merged with other sensory ontologies, whereas O_z represents the remainder of the ontology, i.e. the part not involving in the identified difference. The distinction between $O_{y,i}$ and O_z is particularly important during repair, because we want to control the area that a repair can effect.

Suppose a function f measures some property of *stuff*, then Inconstancy is triggered if $f(\mathit{stuff})$ is predicted to be independent of $b(v)$, but the return value of $f(\mathit{stuff})$ unexpectedly varies when $b(v)$ returns different values (4.46, 4.47, 4.48). We call *stuff* the *inconstancy*, as it is the part of the term that is unexpectedly inconstant, and call $b(v)$ the *variad*, as it is responsible for the unexpected variation in $f(\mathit{stuff})$; the inconstancy might, for instance, be the gravitational constant G and the variad might be the acceleration of an orbiting star due to the gravity, which is suggested by MOND (§4.5.1). The resolution of the detected conflict is to retain all O_x-axioms in the repaired O_x, $\nu(O_x)$, except for the replacement of the occurrences of old *stuff* with $\nu(stuff)(y)$, where y is a new constant, and the replacement of the definition of *stuff* by a new definition of $\nu(stuff)$ in $\nu(O_x)$ (4.50, 4.51) – the new definition establishes a relationship between the variad $b(v)$ and the inconstancy *stuff* (4.49). The variable y is needed because the value to the argument of $\nu(stuff)$ in O_x is not known. The repair also retains all $O_{y,i}(b_i(v) = v_i)$-axioms in the repaired $O_{y,i}(b_i(v) = v_i)$, $\nu(O_{y,i}(b_i(v) = v_i))$, but with all occurrences of old *stuff* replaced by $\nu(stuff)(b_i)$. O_z is repaired in a similar fashion to O_x by replacing occurrences of old *stuff* with $\nu(stuff)(y)$, where y is a new constant. Because axioms in O_z are shared with other sensory ontologies, by definition, the argument of $\nu(stuff)$ in O_z is a new constant rather than a particular b.

4.5.3 Discussion

4.5.3.1 Identification of the Inconstancy

In (4.47), the return values of $f(stuff)$ are derived under the different conditions specified by the corresponding sensory ontology O_2. As already described, Inconstancy detects an unexpected variation only if the return values of $f(stuff)$ varies between at least two ontologies. However, *stuff*, not $f(stuff)$, is regarded to be the inconstancy, even though the detected variation is found in the return values of $f(stuff)$. It may not be immediately clear that an unexpected variation in *stuff* is actually sufficient to cause a variation in $f(stuff)$. Since f takes *stuff* as an argument, treating *stuff* as the inconstancy is the same as treating an argument of some function as the inconstancy if its return values unexpectedly vary. To put this into perspective, suppose a unary function $Mass(x)$ calculates the mass of an object x and $Mass(Rocket)$ returns the mass of a rocket, $Rocket$. Suppose a situation in which the amount of fuel in $Rocket$, and thus its mass, depletes over two snapshots in time, then the values of $Mass(Rocket)$ consequently vary over those periods due to the depletion.. Given the situation, there are various plausible ways to resolve the contradiction; for instance, repair $Mass$ by giving it time as a new argument or, instead, transform $Rocket$ into a function that depends on time.

Suppose that different sensory ontologies give distinct values for $f(\mathit{stuff})$ in different circumstances. Suppose $b(v)$, where b contains variables distinguishing between these circumstances, returns distinct values in each of these circumstances, but is *not* one of the parameters in $f(\mathit{stuff})$, i.e., $f(\mathit{stuff})$ does not depend on $b(v)$. We will call *stuff* the *inconstancy* and $b(v)$ the *variad*.

Trigger: If $f(\mathit{stuff})$ is measured to take different values in different circumstances, then the following trigger formulae will be matched.

$$\exists \tau, \tau', \tau'', \tau''' : Types,\ f{:}\tau \Rightarrow \tau',\ \mathit{stuff}{:}\tau,\ c, c_1, \ldots, c_n{:}\tau', v_1, \ldots, v_n{:}\tau''',$$
$$b_1, \ldots, b_n{:}\tau'', v : \tau'' \Rightarrow \tau''', n > 1, O_x, O_{y,1} \ldots O_{y,n}, O_z{:}\mathbb{O}.$$

$$O_x \vdash \mathit{stuff} ::= c \wedge \tag{4.46}$$
$$O_{y,1}(b_1(v) = v_1) \oplus O_z \vdash f(\mathit{stuff}) = c_1 \wedge \tag{4.47}$$
$$\vdots \quad \vdots \quad \vdots$$
$$O_{y,n}(b_n(v) = v_n) \oplus O_z \vdash f(\mathit{stuff}) = c_n \wedge$$
$$\exists i \neq j \leq n.\ O_x \vdash c_i \neq c_j \vee c \neq c_i \tag{4.48}$$

where the merge $O_{y,i}(b_i(v) = v_i) \oplus O_z$ is the ontology containing observations made under the condition that $b_i(v) = v_i$. The $O_{y,i}(b_i(v) = v_i)$ component represents the maximal part of the ontology that is not merged with other sensory ontologies, whereas O_z represents the remainder of the ontology.

Repair: The repair is to change the signature of all the ontologies to relate the inconstancy, *stuff*, to the variad, $b_i(v)$:

$$\nu(\mathit{stuff}) \quad ::= \quad \lambda y.\ F(c, y(v)) \tag{4.49}$$

where F is a new function, whose value we can seek to determine by regression against the data from the sensory ontologies.

Let $\nu(O_x)$, $\nu(O_{y,i})$ and $\nu(O_z)$ be the repaired ontologies. We calculate the axioms of the new ontologies in terms of those of the old as follows:

$$Ax(\nu(O_x)) ::= \{\phi\{\mathit{stuff}/\nu(\mathit{stuff})(y)\} \mid \phi \in Ax(O_{y,i})\} \setminus \tag{4.50}$$
$$\{\mathit{stuff} ::= c\} \cup \{\nu(\mathit{stuff}) ::= \lambda y.\ F(c, y(v))\}$$
$$Ax(\nu(O_{y,i}(b_i(v) = v_i))) ::= \tag{4.51}$$
$$\{\phi\{\mathit{stuff}/\nu(\mathit{stuff})(b_i)\} \mid \phi \in Ax(O_{y,i}(b_i(v) = v_i))\}$$
$$Ax(\nu(O_z)) ::= \{\phi\{\mathit{stuff}/\nu(\mathit{stuff})(y)\} \mid \phi \in Ax(O_z)\} \tag{4.52}$$

where y is a new constant.

Figure 4.5: The Inconstancy ontology repair plan

4.5.3.2 Condition Vectors

The condition vectors $\vec{cond}_1$ and $\vec{cond}_2$ specify only the conditions that have changed over the two snapshots. Thus, we assume that the modeller determines the relevant conditions that should be specified in the vectors[5].

4.5.3.3 Relationship with *Where's My Stuff*

Although Inconstancy is designed to detect and repair unexpected variations in some measured property of *stuff*, it actually shares underpinnings with WMS. Suppose a situation involving an unexpected variation such that the Inconstancy trigger formulae (4.46, 4.47, 4.48) can be instantiated with the following substitution[6]

$$\{\, O_x/A,\ O_y/B,\ v/w,\ b_i/a_i,\ v_i/w_i,\ f/\lambda x.\,x,\ stuff/n,\ c/p,\ c_i/q_i \,\}$$

which gives rise to the following setup

$$\begin{aligned} A &\vdash n ::= p \wedge && (4.53)\\ B(w(a_1) = w_1) &\vdash n = q_1 \wedge && (4.54)\\ \vdots \quad &\ \vdots \quad \vdots && \\ B(w(a_n) = w_n) &\vdash n = q_n \wedge && \\ \exists i \neq j \leq n.\, A &\vdash q_i \neq q_j \vee p \neq q_i && (4.55) \end{aligned}$$

thus, some constant n is predicted to have a value p, but is observed to have different values q_i under different conditions, which consequently gives rise to an unexpected variation. Given this setup, the WMS trigger formulae could in fact be instantiated, provided that an order rather than a mere inequality between p and q_i can be derived (4.55), e.g. $p > q_i$ or $p < q_i$ for some $i \leq n$. Suppose $p > q_1$. The WMS trigger can then be instantiated with the substitution:

$$\{O_x/A,\ O_y/B(w(a_1) = w_1),\ f/\lambda x.\,x,\ stuff/n,\ v/q_1\} \qquad (4.56)$$

where only ontologies A and $B(w(a_1) = w_1)$ are within the scope of repair, whereas the remaining ontologies, i.e. $\forall 1 < i \leq n.\ B(w(a_i) = w_i)$, are ignored. Thus, following

[5] We have avoided the obstacles imposed by the *frame problem*, outlined by McCarthy and Hayes (1969); the frame problem is a special case of the problem of complete description (Van Brakel (1992)). The application or extension of a well-studied formalism, such as the formalisms by Reiter (1979) and McCarthy (1980), is beyond the scope of the research, so the selection of relevant conditions is determined by the user outside of the system.

[6] We exclude O_z to simplify the discussion.

the WMS repair rules, (4.18, 4.19), n is repaired by the introduction of invisible *stuff* in A[7], i.e.

$$\nu(A) \vdash n ::= p \tag{4.57}$$

$$\vdash n_{invis} ::= n - n_{vis} \tag{4.58}$$

$$\nu(B(w(a_1) = w_1)) \vdash n_{vis} = q_1 \tag{4.59}$$

which resolves the *detected* contradiction. However, if $q_i \neq n$ for some $i > 1$, then a contradiction still remains between $B(w(a_i) = w_i)$ and A. So, a single application of WMS is not sufficient if the sensory values of n are different. Inconstancy handles this kind of fault appropriately, because it is designed to analyse multiple sensory ontologies, whilst WMS works with only one. That said, WMS can be applied again on $\nu(A)$ and $B(w(a_i) = w_i)$ if a contradiction remains. However, even the ultimate repair makes no indication that n is supposed to depend on a new parameter, but instead it introduces one or more additive terms representing various invisible stuffs, which act as offset values. In general, both Inconstancy and WMS[8] are able to, at least eventually, *resolve* the contradiction arising from unexpected variations. Overall, Inconstancy gives a more appropriate repair to the problem if $\exists i \neq j \leq n.q_i \neq q_j$, since the repair involves explicitly increasing the arity of *stuff* by one and adding a new dependency to *stuff* rather than introducing a term that does not necessarily vary with a hidden dependency. WMS gives a more appropriate repair if $\forall i \neq j \leq n.q_i = q_j$, because it is not clear that there is an unexpected *variation* in the value of *stuff* according to the sensory data.

4.6 The *Unite* Ontology Repair Plan

Since a key objective of ORPs is to resolve all common kinds of faults in ontologies, ORPs are not designed only to resolve logical contradictions between ontologies, but also to provide them with extra logical power, e.g., solving a problem of underspecification (Definition 7, p. 34). The strategy to increasing the strength must not be arbitrary, e.g., not adding arbitrary axioms that are consistent with the ontology, but that the newly derivable sentences should be consequences of both the ontologies and implications for the natural interpretation of Physics. An example conflict is that, given two different terms referring to the supposedly the same concept or object, neither the equality nor the inequality between the two terms is derivable from the theory. Note

[7]If $A \vdash p < q_i$, then the introduction is made in B instead.

[8]Applicable only if a partial order between the return values of *stuff* can be derived.

that the failure to derive the equality does not necessarily mean that the inequality is necessarily derivable, because we adopt the open-world semantics, which assumes that the derivability and the truth of a statement are independent of each other. A natural way to resolving such conflict is to insert a new axiom that equates the two terms.

4.6.1 Motivating Example: The Morning and Evening Stars

Because Venus is the brightest celestial object in the sky after the Sun and the Moon, it has been observed since ancient times. Venus, being nearer to the Sun than the Earth is, can never be far from the Sun when viewed from Earth. Before the Sun rises, Venus might be near the Sun in the east, as a *morning star*. After the Sun sets, it might also be near the Sun but in the west, as an *evening star*. These appearances in the sky had led Ancient Egyptians and Ancient Greeks to believe that they were of two distinct objects in the sky. It was only by identifying that the 'two' 'stars' follow the same orbit, and thus share the same value for the defining property, that they were understood to be appearances of the same object. Suppose *MSESTheory* and *Obs* are, respectively, ontologies containing the state of a theory about the distinction between the morning star, *MS*, and the evening star, *ES*, and the sensory information required to compute their orbits, *Orbit*. We can represent the original ontologies as follows:

$$MSESTheory \nvdash MS = ES \tag{4.60}$$

$$Obs \vdash Orbit(MS) = Orbit(ES) \tag{4.61}$$

where (4.60) mean that the morning and evening stars cannot be inferred to be the same according to *MSESTheory*.

The required repair in order to resolve the fault is to equate *MS* with *ES* (Bundy, 2009).

$$\nu(MSESTheory) \vdash MS = ES \tag{4.62}$$

where $\nu(MSESTheory)$ denotes the repaired *MSESTheory*. This episode highlights several key issues about detecting neither the equality nor the inequality between supposedly the same thing and repairing it by equating them. For instance, the equality between *MS* and *ES* is deemed to be the appropriate repair because the fact that they follow the same orbit. Moreover, the mere insertion of an axiom that asserts the equality between *MS* and *ES* is sufficient because it enriches the theory. However, $\nu(MSESTheory)$ may not be necessarily consistent, because $MS \neq ES$ might be a theorem of *MSESTheory*[9]. So, the new axiom might create a new logical conflict. That

[9]Many Ancient Greeks believed that the morning and evening stars were actually two distinct stars.

Suppose that $stuff_1$ and $stuff_2$ are of the same type, τ, but have different names in one ontology, O_1. Suppose the function *dp* measures the defining property for all instances of the type τ and returning a value of type τ'. From another ontology, O_2, both $stuff_1$ and $stuff_2$ are deduced to take similar values for the function *dp*.

Trigger: If neither the equality nor inequality between $f(stuff_1)$ and $f(stuff_2)$ is deducible from O_1 and both $stuff_1$ and $stuff_2$ share a defining property of their type, *dp*, then the following formula will be triggered:

$$O_1 \nvdash f(stuff_1) = f(stuff_2) \wedge \tag{4.63}$$

$$O_M \vdash DefProp(dp, \tau) \wedge \tag{4.64}$$

$$O_2 \vdash dp(stuff_1) = dp(stuff_2) \tag{4.65}$$

where O_M is a meta-level ontology; $DefProp(dp, \tau)$ means that function *dp* measures a defining property of objects of type τ returns true if the values of the defining property of two objects are the same.

Repair: The repair is to add an equality between $f(stuff_1)$ and $f(stuff_2)$ as a new axiom to O_1. O_2 is unchanged. Let $\nu(O_1)$ and $\nu(O_2)$ be the repaired ontologies. We revise the axioms for the new ontologies in terms of those of the old as follows:

$$Ax(\nu(O_1)) ::= \{ f(stuff_1) = f(stuff_2) \} \cup Ax(O_1) \tag{4.66}$$

where $Ax(O)$ is the set of axioms of ontology O. The axioms of $\nu(O_1)$ are the same as those of O_1, except for the addition of the new definition.

Figure 4.6: The Unite ontology repair plan

said, it is still beneficial to insert the new axiom, as it essentially transformed the fault from under-specification to over-specification. If further repairs were required, other repair techniques could be useful, e.g., belief revision.

4.6.2 Overview of Unite

The *Unite* ORP (Figure 4.6) is inspired by the morning and evening star example, described in §4.6.1. It detects that the equality between two different terms referring to supposedly the same concept or object is not derivable from the theory and repairs it by strengthening it. Unite, therefore, is not driven by a contradiction, unlike the previous three ORPs. For two terms to refer to the same thing, each of their features and properties should take the same value, which is the principle behind Leibniz's equality. As long as one of the terms yields a different value for one of the properties, the two terms should be considered to refer to different things. By this principle, we may need to examine all possible properties in order to conclude that two terms refer to the same thing. Unfortunately, explicitly indicating all possible properties and their values is an impractical task, because it may be too immense for the available resources and, more importantly, not every property about an entity is actually known in practice. Moreover, not all properties are sufficient, e.g., colour. This leads to the need for determining a manageable set of properties that are themselves sufficient and necessary for deciding equality between instances of a particular type, which we call *defining properties*. The defining properties for a particular type are typically a small subset of possible properties, but they, collectively or individually, alone are sufficient. For instance, a defining property of celestial objects is their orbits defined in a 4-dimensional space, based on the fact that no two solid objects can occupy the same space at the same time. We will define that the function *dp* is a defining property for the type τ using the representation $DefProp(dp, \tau)$ (4.64). Since *dp* is a function, it can be compounded in cases that require multiple defining properties. For instance, quantum objects are defined by quantum states. The energy level of a quantum object is only part of a quantum state, so to define a quantum state completely, we need to include other properties, such as the position and the spin.

Beside checking that the two terms yield the same value for the defining property (4.65), the equality between them should not already be derivable (4.63). If the two terms are already deemed to be equal, then there is no need for repair as there is no fault to be repaired and nothing to gain from the information provided by the other ontology.

The repair to such conflict is to simply insert the equality between the two terms as a new axiom.

4.6.3 Discussion

Unite is triggered if equality or inequality between $f(stuff_1)$ and $f(stuff_2)$ cannot be proven rather than between $stuff_1$ and $stuff_2$. Making $stuff_1$ and $stuff_2$ arguments of a function f increases the generality of the ORP, as $stuff_1$ and $stuff_2$ can be instantiated to a subterm. Because $stuff_1$ and $stuff_2$ are arguments of f, if they are regarded to be equal, then $f(\mathit{stuff}_1)$ and $f(\mathit{stuff}_2)$ must also be equal due to f being deterministic, i.e. $\mathit{stuff}_1 = \mathit{stuff}_2 \longrightarrow f(\mathit{stuff}_1) = f(\mathit{stuff}_2)$.

Ensuring that (4.63) hold by explicitly showing the unprovability of an equality between $f(\mathit{stuff}_1)$ and $f(\mathit{stuff}_2)$ is undecidable, as validity in HOL is undecidable. To match (4.63) in practice, the user could assert the unprovability and guarantee that the sentence is indeed not provable in the ontology. Alternatively, it could be reformulated so that the unprovability must be verified as part of a proof obligation. The relevant proof obligation requires that the inequality between $f(\mathit{stuff}_1)$ and $f(\mathit{stuff}_2)$ is provable in one consistent extension of O_1 and that the equality between $f(\mathit{stuff}_1)$ and $f(\mathit{stuff}_2)$ is provable in another consistent extension. More formally,

$$O_1{}' \vdash f(\mathit{stuff}_1) \neq f(\mathit{stuff}_2) \quad (4.67)$$

$$O_1{}'' \vdash f(\mathit{stuff}_1) = f(\mathit{stuff}_2) \quad (4.68)$$

where O_1' and O_1'' are different extensions of O_1. Although this formulation implies the unprovabilities at the meta-level, it requires a considerably stronger model of the domain of discourse under consideration than (4.63) needs, because two additional ontologies that are extensions of O_1 are presumed to exist within the representation.

the

4.7 The *Spectrum* Ontology Repair Plan

Unary predicates are often used to express that a certain individual satisfies some particular property, e.g., $\mathit{Red}(\mathit{ball})$ and $\mathit{Yellow}(\mathit{car})$ typically mean that the object *ball* has the colour red and the object *car* has the colour yellow, respectively. Unary predicates can be thought of as set memberships in a way that the name of the predicate corresponds to the name of a set and the predicate is true if and only if the argument is a member of the set. This is often sufficient for simple and controlled problem domains, but is a relatively unnatural representation for domains involving complex relationships between concepts, such as Physics. The set representation is popularly used in knowledge representation because of its close relationship with the semantics of DLs. Since we are working in HOL, we can utilise its power to allow ontologies to have a more natural and expressive representation.

4.7.1 Motivating Example

It was with Maxwell's equations that electricity, magnetism and light were related, thereby unifying the previously separate fields of electromagnetism and optics and predicting the nature of electromagnetic (EM) waves. EM waves are characterised by the respective wavelength, which can be plotted on a graph to produce a spectrum. Classes of EM radiation can be created by segmenting the spectrum into, e.g., radio, microwave, visible, etc. Suppose ontology *EMTest* contains experimental data about the colour of various light beams.

$$EMTest \vdash Red(beam1) \tag{4.69}$$

$$EMTest \vdash Green(beam2) \tag{4.70}$$

$$EMTest \vdash Blue(beam3) \tag{4.71}$$

where *Red*, *Green* and *Blue* are unary predicates that are true if the beam is of the respective colour; $beam1$ - $beam3$ denote various light beams[10]. From (4.69 - 4.71), the unary predicates do not clearly express the relationships between the three beams or between the three predicates. Further, the unary predicate representation is not desirable, particularly if, e.g., the concept of colour was more refined and captured the wavelength dimension, then $Red(beam1)$ would become $650nm(beam1)$ and $Green(beam2)$ would become $510nm(beam2)$, etc. In this case, since we know the colour spectrum is continuous, many sets might be required in the representation where each set corresponds to a particular wave length. It would, therefore, be more preferable to use a function that takes an object as an argument and returns a colour as value. Thus, an alternative representation is to treat the predicate names as values:

$$EMTest \vdash Colour(beam1) = Red \tag{4.72}$$

$$EMTest \vdash Colour(beam2) = Green \tag{4.73}$$

$$EMTest \vdash Colour(beam3) = Blue \tag{4.74}$$

where *Colour* is a function taking a light beam as argument and returning its colour. Because *Colour* is a function, it guarantees that each beam can have at most one colour. If we work with composite colour mixtures, then the output of the *Colour* function could be a triple instead, representing RGB distributions; for instance, $Colour(beam1) = (255, 0, 0)$. Another motivating example is the unified theory of electro-magnetic radiation, as studied by Bundy (2010).

[10]Note that the colours of the visible range can be represented as a combination of red, blue and green is a feature of physiology of the human eye.

Suppose $\mathcal{Q}$ is the set of objects and $\mathcal{P}$ is the set of unary predicates in some ontology O.

Trigger: If there exists exactly one unary predicate p in the non-singleton set of all unary predicates $\mathcal{P}$ that is true for all objects o in the non-singleton set of objects $\mathcal{Q}$, then the following formula will be triggered:

$$|\mathcal{Q}| > 1 \ \wedge \ |\mathcal{P}| > 1 \ \wedge \ \forall o \in \mathcal{Q}. \ \exists! p \in \mathcal{P}. \ p(o) \tag{4.75}$$

where $\exists! p$ is the uniqueness quantification over p, which means that there is one and only one such p; the term $|\mathcal{Q}| > 1 \ \wedge \ |\mathcal{P}| > 1$ prevents the trivial case, in which there is one or fewer object in $\mathcal{Q}$ and that there is one or fewer predicate in $\mathcal{P}$.

Repair: Let $\nu(O)$ be the repaired ontologies. We define the signature and axioms of $\nu(O)$ as follows:

$$\begin{aligned} Sig(\nu(O)) \ ::= \ & Sig(O) \setminus \{p : \tau_o \Rightarrow bool \mid p \in \mathcal{P}\} \ \setminus \\ & \{ \ o{:}\tau_o \mid o \in \mathcal{Q} \ \} \cup \{ \ f : \tau \Rightarrow \tau' \ \} \cup \\ & \{ \ o{:}\tau \mid o \in \mathcal{Q} \ \} \cup \{ \ p{:}\tau' | p \in \mathcal{P} \ \} \end{aligned} \tag{4.76}$$

$$Ax(\nu(O)) \ ::= \ Ax(O) \cup \{ \ p(o)/f(o) = p \mid p \in \mathcal{P} \wedge o \in \mathcal{Q} \ \} \tag{4.77}$$

where $Sig(\nu(O))$ is the set of signature declarations of $\nu(O)$; $Ax(\nu(O))$ is the set of axioms of $\nu(O)$; τ_p is the input type of each unary predicate; and, τ_o is the type of each object.

definition; the axioms of $\nu(O_2)$ are the same as those of O_2 except for the renaming of the original stuff to the visible stuff.

Figure 4.7: The Spectrum ontology repair plan

4.7.2 Overview

The *Spectrum* ORP detects that there exists some unary predicates in the ontology and is a step toward designing a repair plan that constructs spectrums. Like Unite, Spectrum is not driven by a contradiction, so triggering Spectrum on some ontology does not mean that it is inconsistent, but instead the representation can be enriched. The ontology is repaired by changing the representation from multiple unary predicates into a single unary function that takes the same input as the original unary predicates but has the names of the predicates in its range. To avoid trivial matches, (4.75) restricts on the properties of $\mathcal{Q}$, the set of objects which are arguments of the unary predicates, and $\mathcal{P}$, the set of unary predicates in the ontology. The term $|Q| > 1$ in (4.75) requires that there are at least two elements in the set of objects, which prevents the case where f, which is the function created by the repair to resolve the fault, has a domain that is a singleton. The term $|\mathcal{P}| > 1$ in (4.75) requires that there are at least two elements in the set of unary predicates, so that f is not a constant function and its output depends on its argument. If there are more than one predicate that is true for some object, then the new function would not be deterministic. Spectrum repairs the ontology by retracting the declarations of the unary predicates and inserting a declaration of the new function, f, that takes objects as arguments and recycles the names of the original predicates as its return values (4.76). For each object o, the value of $f(o)$ is asserted to be the name of the original predicate that was true for o (4.77). So, all occurrences of $p(o)$ in the old axioms will be replaced by $f(o) = p$.

4.7.3 Discussion

Spectrum and Unite actually share a common principle, which is that seemingly the same concepts should be related. For Spectrum, the names of the predicates that are true together make up the range of a new function; for Unite, things that share the same defining property are equated and merged into one. Unfortunately, Unite cannot be adapted to create a spectrum in its repair. Unite requires that the manifestations of *stuff* should originally be of the same type and that they already share the same defining property, but it does not capture the values that could produce a spectrum. Spectrum, on the other hand, does not verify the physical relationship between the measured properties, but creates a spectrum corresponding to a range of valid values.

4.8 Summary

This chapter describes the key mechanism for realising and increasing the automation of ontology evolution in Physics: ontology repair plans. ORPs are designed to resolve ontological faults that arise from a merge of multiple globally inconsistent ontologies, i.e. over-specified ontologies, or under-specified ontologies. A detected fault in an ontology or between ontologies, therefore, does not necessarily require a contradiction, but might be that the representation can be enriched in order to enhance the inference capability. Each ORP is presented by providing the corresponding trigger formulae, which define the unique type of faults handled by the ORP, and repair rules, which define the transformations required in order to resolve the conflict or enrich an ontology. The trigger formulae and repair rules of an ORP can be, respectively, viewed as the precondition and effect of an ORP, because the trigger formulae must be satisfied in order for an ORP to produce the relevant repair. The existential quantification over ontologies, e.g., O_1 and O_2, and functions, e.g., f and *stuff*, means that at least second-order logic, is required for reasoning. Quantification over ontologies permits a high generality in each ORP and is particularly useful in domains containing many ontologies/theories, e.g., natural sciences and general real-world semantics.

Each ORP is designed to tackle a unique kind of fault and performs appropriate repair operations[11], as illustrated in Figure 4.1. All of the ORPs are listed, with their requirements and types of repair operations. The requirements imposed by the ORPs are:

- Minimum ontologies: The minimum number of ontologies in order to trigger the ORP.
- Over-specification driven: Whether the ORP detects a logical contradiction, which arises from the merge of multiple ontologies.
- Under-specification driven: Whether the ORP detects a weakness in the theorems, which can be enriched.

The repair operations can be classified into five kinds:

- Axiom insertion: Adds a new axiom into an ontology.
- Axiom retraction: Deletes an axiom from an ontology.
- Signature creation: Declares a new signature element.
- Signature retraction: Deletes a signature element from the language.
- Abstract concept projection: Creates a new concept that does not correspond to an existing concept in the input ontologies or is undefined.

[11] Although certain ontological setups can be repaired by more than one ORP, the corresponding repaired ontologies yield substantially different theorems.

	WMS	Reidealisation	Inconstancy	Unite	Spectrum
Minimum ontologies	2	2	3	1	1
Over-specification driven	✓	✓	✓	✗	✗
Under-specification driven	✗	✗	✗	✓	✓
Axiom insertion	✓	✓	✓	✓	✓
Axiom retraction	✓	✓	✓	✗	✓
Signature creation	✓	✓	✓	✗	✓
Signature retraction	✓	✓	✓	✗	✓
Abstract concept projection	✗	✗	✓	✗	✗

Table 4.1: Requirements of and kinds of repair performed by ontology repair plans.

Because both Unite and Spectrum are designed to handle under-specified ontologies, they are the only ORPs that can be triggered by only one ontology. The trigger formulae of Unite consider two ontologies merely for the purpose of better fitting the ORP to the Physics domain, with O_1 being the predictive ontology and O_2 being the sensory ontology. The two ontologies can be combined and no contradiction will necessarily arise. Spectrum is designed to enrich the representation of an ontology without acquiring new information from external sources, so it inherently considers one ontology. Both WMS and Reidealisation require two ontologies such that the merge of which gives rise to a contradiction. Inconstancy, on the other hand, requires at least three ontologies, as a variation in some values can be depicted only with at least two data points.

Abstract concept projection is a kind of operation that results in some new term being conjectured by the repair operation without corresponding to an existing symbol or a definition; that is, the ORP decides that it is essential to create a new concept in the domain of discourse in order to resolve the detected fault. Inconstancy is the only ORP that projects abstract concepts as it invents a new constant y as part of the repair (4.50). The new constant is needed because of the fact that the arity of *stuff* is incremented, but the value the new argument should take is not known in O_1. Inconstancy also invents F, but it is not an abstract concept as it can be determined by regression against the sensory information. Both WMS and Spectrum also invent concepts, but they are not abstract. For instance, WMS invents the symbol $stuff_{invis}$ and defines it to be the difference between *stuff* and $stuff_{vis}$. If the values of *stuff* and $stuff_{vis}$ are known, then $stuff_{invis}$ can be computed. Spectrum invents a new unary function F and defines its range to be the names of the old unary predicates.

Considering all five kinds of repair, the repair performed by Inconstancy is the most diverse among all ORPs, as it covers all five kinds of repair. We consider Inconstancy to be the most complex of all; for example, it measures an unexpected variation of some value by inspecting for a change in a hidden variable and the repair behaviour depends on whether the part of the ontology is shared by other sensory ontologies or not. The least diverse ORPs, in contrast, is Unite, which performs only one kind of repair: an insertion of an axiom stating that the two stuffs are equal.

As we have seen, the ORPs utilises a various combinations of repair kinds. This illustrates a key benefit of using HOL for ontology evolution, enabling the flexible composition of operations for signature and axiom manipulations. Moreover, HOL's polymorphism of variables and other symbols permits the generality of the repair plans for evolution and their applicability over diverse cases, which will be investigated in later chapters. Indeed, the ontological faults we have addressed are not exhaustive and the design of new ORPs is a promising avenue for further work.

Chapter 5

Overview of the GALILEO System

5.1 Introduction

In this chapter we give an overview of the GALILEO system (Chan et al., 2011; Lehmann et al., 2011; Chan and Bundy, 2009; Lehmann et al., 2012), which contains an implementation of the ORPs described in Chapter 4. In §5.2, we review the research objectives specific to the implementation of the GALILEO system and the way limitations can be minimised. Next, in §5.3, we describe the scope of the research into building the system. In §5.4, we present the architecture of the system, introduce the key components, and describe the flow of the system.

5.2 Research Objectives

The overall aim of this project is to demonstrate that ontology evolution in Physics can be mechanised by implementing the ORPs designed within a system, in which if the input Physics ontologies contain a recognisable fault at the logical level or at the representational level, then the system is capable of performing diagnosis and repair. A logical conflict is a type of conflict that results in an inconsistent merge of the given ontologies and a representational conflict is one that is caused by incompatible, undesirable or unoptimised ontological representations of concepts. Our main interest is, therefore, in mechanising the repair of various kinds of over- and under-specifications (§9, p.34). To constrict the space of possible repair methods, a method of diagnosis is designed to establish the conflict that causes the fault. The result from diagnosis helps the system select the appropriate repair strategy in order to remove the conflict from the ontologies. An appropriate repair strategy must carefully control the set of derivable sentences in the repaired ontologies. We do not want to adopt naive repair operations, such as a complete axiom retraction, which can theoretically always eliminate certain faults, e.g., over-specifications, in ontologies.

At the crux of the GALILEO system is an implementation of the ORPs. As presented in Chapter 4, each ORP contains a trigger pattern that represents that a certain kind of fault exists in the given ontologies and a set of transformation rules that specify the characteristics of new ontologies that do not contain the detected conflict. Our system provides facilities for performing ontological conflict diagnosis and ontology repair that imitate the theoretical behaviour of each ORP. For each ORP, we must consider the ramifications of repairing the detected conflict w.r.t. the rest of the ontology, because a repair should be prohibited from producing new faults that require other alterations. It is, therefore, crucial to have an understanding of the repair semantics underlying the ORPs and investigate their direct and implied effects on general ontological configurations, e.g., ontologies that do not share a signature and ontologies that depend on others.

Fault diagnosis requires deep reasoning about the ontologies, but since reasoning in HOL is undecidable, the level of automation in ontology evolution is inherently limited. A key objective of research is to build a system that mechanises the ontology repair process of higher-order ontologies with as much automation in the ontology repair process as possible. User interaction is used to guide the search when complexity is beyond automation. Repair may take place only if the input ontologies satisfy the diagnostic requirements described in the trigger formula of at least one ORP. The end result is that meta-level analysis and reasoning is fully automated, whilst object-level reasoning requires interaction. Some degree of interaction in object-level reasoning is almost inevitable, because of the undecidability of HOL. Increasing the automation of reasoning is beyond the scope of current research.

In the case of a successful repair, the output contains the repaired ontologies with their new signatures and/or new axioms, depending on the ORP invoked. The modelling approach adopted to represent the original ontologies determines the complexity of repair. If the original ontologies are part of a network, in which some ontologies may be merged with others, then the repair procedure must compute the new dependencies and manage the propagation of repair. If this kind of setup is given as input, the dependencies among the repaired ontologies should imitate those among the original ontologies.

5.3 Scope of Research

The scope of the project covers a wide area of research topics, each with its own difficult and unsolved problems. For example, improving the automation of higher-order reasoning for general problems has been proven to be an immense challenge for the automated reasoning community. Moreover, the philosophical foundation of the evolution of multiple ontologies as an epistemological approach to defeasible reasoning has not yet been established. It, therefore, further complicates the problem of increasing the generality of our approach and scaling the system to handle a diversity of ontological configurations, e.g., from a 'big' ontology explicitly declaring all the signature elements and containing all the axioms to 'small' ontologies that import signature elements and axioms from others through combinations. Although we introduce highly general techniques for evolving ontologies, we do not provide an account of the underlying epistemological issues. Thus, the focus of attention needs to be on striking a balance among understanding and addressing these issues, realising ontology evolution in HOL and reducing the complexity of the evolution process.

Our primary interest specific to the development of the system is in studying the reasoning requirements of ontology evolution and providing the system with reasoning capabilities. Our focus is on augmenting Isabelle with the functionality of diagnosing and repairing ontologies. As Isabelle is a theorem prover and is not designed for ontology evolution, enabling ontology evolution in Isabelle requires an unusual use of it and a substantial extension. This thesis shows how some of the fundamental limitations can be overcome and that GALILEO is a first attempt to develop a general ontology evolution system.

5.4 Design and Architecture

The GALILEO system consists of two major components: the *diagnosis engine* and the *repair engine*. At the centre of the interaction is the user, who provides the input ontologies and the needed interactive guidance in order to fully proceed with diagnosis and invoke the automated repair. Isabelle is used for performing all reasoning tasks – its higher-order matching algorithm is used for both the implementations of diagnosis and

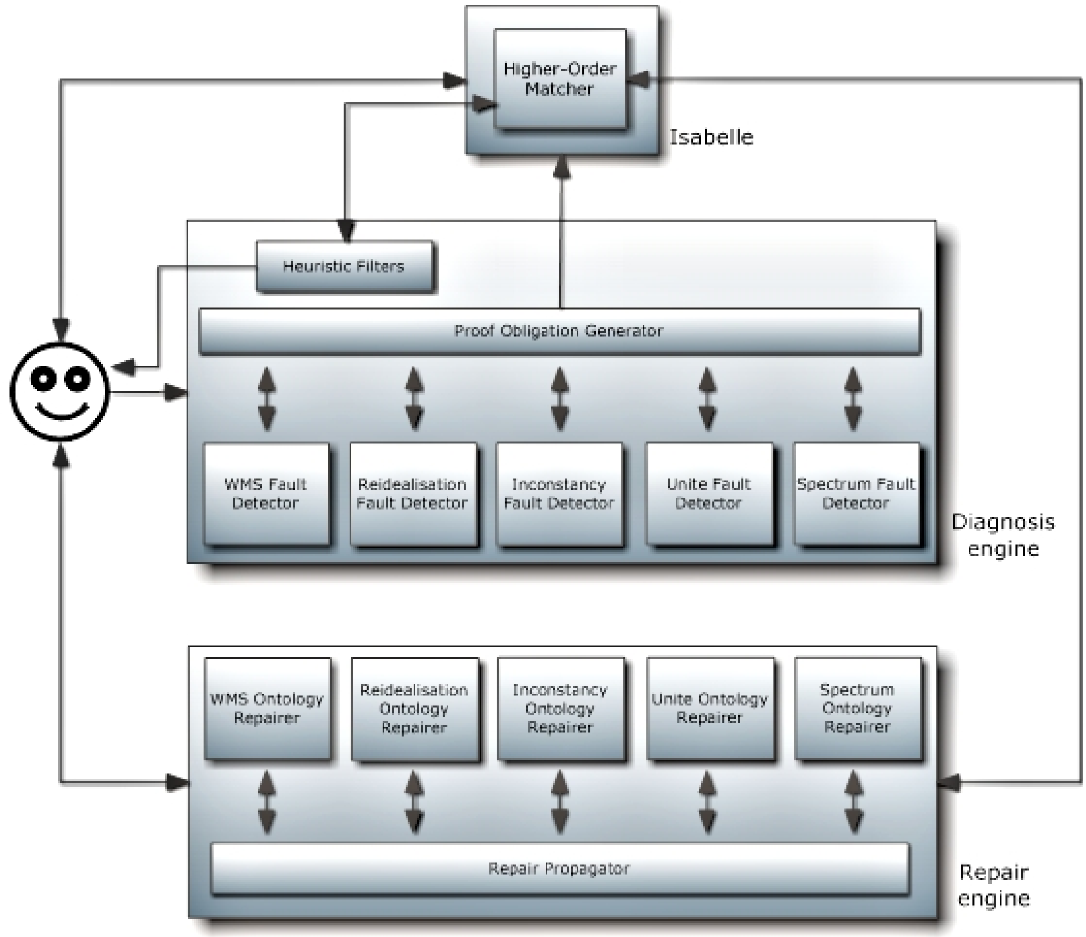


Figure 5.1: High-level architecture and interactions between key components in GALILEO.

repair. Figure 5.1 depicts the high-level block diagram of the architecture of GALILEO and the interactions between both the user and the system and among the components within the system. Since GALILEO is implemented as an extension of Isabelle, it is loaded along with the core of Isabelle. The user first supplies the input ontologies in an Isabelle theory file and Isabelle processes the file in the standard manner. The user then initiates the ontology evolution process by invoking a particular ORP and supplying the relevant preparation information, which is further explained in Chapter 6. One ORP is invoked at a time, because the proof for ensuring that the input ontologies contains the corresponding fault typically requires some user interactions. Ideally, if the proofs were completely automated, then all ORPs would be invoked simultaneously rather than invoking one ORP at a time. Of course, theoretically, all ORPs could still be invoked simultaneously even when user interactions are required to help find proofs, but we believe users may be overwhelmed by the execution of multiple ORPs. The fault detector of the ORP will then attempt to instantiate the existential variables in the trigger formula and generate a proof obligation. The proof obligation is typically in the shape of the trigger formula of the invoked ORP and ensures that the fault which the ORP is designed to resolve indeed exists in the input ontologies. In most cases, the fault detector of the invoked ORP produces multiple plausible diagnoses. The ontology evolution process is blocked until the proof obligation is discharged, i.e. a proof is found. Once it is discharged, the diagnosis engine utilises Isabelle to produce a collection of logically valid and physically meaningful diagnoses[1]. Because of the typically unmanageable number of diagnoses, several heuristics for filtering the physically meaningful diagnoses are applied. If a diagnosis is deemed to be meaningful (Chapter 6), then it is returned to the user. In most cases multiple diagnoses are returned, where each corresponds to a physically plausible repair, so the user needs to select a diagnosis that fits with the current problem context and application. The repair engine reads the selected diagnosis and extracts from it information relevant to the execution of the transformation of the signature and/or axioms of the input ontologies. Not every axiom in the input ontologies is subject to repair, so higher-order matching is applied to identify the axioms that require transformation. Using the matching results, the input ontologies are repaired according to the repair rules of the ORP invoked and the repaired ontologies are then internally represented. If the ontologies under consideration depend on additional ontologies, then the effect of the repair may be propagated to these ontologies as well. In more complex situations, the evolution of ontologies is iterative, so the repaired ontologies themselves may be subject to further evolution by invoking an ORP on them as well.

5.5 Modelling and Reasoning about Physics

Unlike the controlled domains typically studied in ontology research, such as a shop or a library, the Physics domain deals with highly general concepts that are intended to be applicable across a diverse range of scientific settings. We discuss in this section the importance of employing higher-order logic for both producing natural and general models of Physics concepts and for reasoning about their properties.

[1]As to be discussed in Chapter 6, not all logically valid diagnoses are physically meaningful, and a separate mechanism is needed to filter interesting ones.

5.5.1 Higher-Order Logic and Physics

Concepts with dynamic properties play a major role in the Physics domain; for instance, the notions of velocity and acceleration of bodies, which are the changes of distance and of velocity over time, respectively, are fundamental to the foundation of Physics. Such concepts are bound to vary over a period, so they are better represented as mathematical functions returning values corresponding to the given input. In order to work with changing physical processes, calculus is essential to the mathematical foundation. Some key measures in calculus, including derivatives and integrals, are naturally represented as higher-order functions, even though they can be embedded in first-order set theory.

As with other more complex aspects of the logical formalisation of Physics, attempts have been made to axiomatise specific Physics theories and their results have shown that it would be desirable for such axiomatisations to be written in logics more expressive than FOL. For instance, although the axiomatisation of Special Theory of Relativity might be within the bounds of FOL (Székely, 2010), more expressivity is needed in the axiomatisation in order to have a general proof of the Twin Paradox and to axiomatise the General Theory (Madarász et al., 2006). In the rest of this section, we will illustrate our argument for the need for HOL with examples that require less specialist Physics knowledge.

5.5.2 Example: Representation of Orbits

Arguably, concepts involving changing quantities could potentially be represented using a representation more basic than functions. Take the orbit of a star as an example. There are appropriate equations for calculating its precise position in space for a given time moment. That said, an orbit could also be perceived as a collection of four dimensional points: three dimensions of space and one of time. However, given the uniqueness of each point, the collection can be accurately modelled only as an infinite set[2]; although it could be approximated finitely, it would still yield an unnatural and excessively complex representation. Further, its continuity and uniqueness of each point given its time would be complicated to formalise. Although the need for function objects in the representation can be met by adopting FOL in general, a natural representation of Physics requires more expressive power in the logic. For instance, we

[2]The countability of the set depends on the assumed quantisation of space and time and, thus, the relevance of Planck's constants.

can provide a concise and general definition to the meaning of an eventual collision between two objects in terms of an intersection of their paths and assert that two stars $star_1$ and $star_2$ eventually collide as:

$$WillCollide(pathA, pathB) ::= \exists t{:}Mom.\ pathA(t) = pathB(t) \tag{5.1}$$

$$WillCollide(orbit(star_1), orbit(star_2)) \tag{5.2}$$

where $WillCollide$ is a boolean function taking two path functions as arguments, and each of *pathA* and *pathB* takes a time moment of type *Mom* as an argument and returns a 3D position in space (5.1). Suppose $orbit(s)$ returns the orbit of an object s and we know the orbits of $star_1$ and $star_2$, (5.2) means that $star_1$ and $star_2$ will eventually collide. Since $WillCollide$ is of type

$$(Mom \Rightarrow Position) \Rightarrow (Mom \Rightarrow Position) \Rightarrow bool \tag{5.3}$$

and is, therefore, it is a second-order term and the logic underlying the representation should be higher-order. The conciseness of representation in the example is a result of adopting an expressive logic.

5.5.3 Example: Representation of Latent Heat

Physics concepts are often intended to be applicable across a diverse collection of entities in which these entities could be of various basic properties, e.g., different substances or states. If the definition of a concept depends on some particular property, a relevant set of equations or table of values is typically available for those satisfying that property. For instance, *latent heat*, which is the kind of heat released or absorbed by a substance during phase change, e.g., the melting of ice or boiling of water, is defined as

$$Q = m \times L \tag{5.4}$$

where Q is the amount of energy released or absorbed, m is the mass of the substance, and L is the specific latent heat. The definition in (5.4) is applicable to both fusion and vaporisation, but the value L takes depends on the substance and the type of phase change undergone; for instance, with water, the specific latent heat of fusion is 334kJ/kg, whereas the specific latent heat of vaporisation is 2260kJ/kg. These values are commonly tabulated in separate tables: one for fusion and one for vaporisation. A formal representation of (5.4) in a multi-sorted FOL ontology O could be

$$\forall o{:}Obj, t{:}Period.\ Heat(o,p) := Mass(o,p) \times L(o,p) \in Ax(O) \tag{5.5}$$

where $Heat(o,p)$ returns the amount of heat an object o is released or absorbed over a period p, $mass(o,p)$ returns the average mass of object o over period p, and $L(o,p)$ returns the specific latent heat of o over p. The value $L(o,p)$ depends on the type of phase change o undergoes over p, so additional assertions about the phase change o undergoes are also required:

$$\{\ \forall o,p.\ IsWater(SubstanceOf(o,p)) \wedge Fusion(o,p) \longrightarrow L(o,p) = 334, \tag{5.6}$$

$$\forall o,p.\ IsWater(SubstanceOf(o,p)) \wedge Vaporisation(o,p) \longrightarrow L(o,p) = 2260 \quad \ldots\} \subset Ax(O) \tag{5.7}$$

where $SubstanceOf(o,p)$ returns the substance of o over period p; $IsWater(s)$ is true if and only if s is water; $Fusion(o,p)$ is true if and only if the o undergoes fusion over period p; and, $Vaporisation(o,p)$ is true if and only if o undergoes vaporisation over p. So, if a symbol $puddle{:}Obj$ represents some puddle of water undergoing fusion at time T_0, then we must have

$$O \vdash IsWater(SubstanceOf(puddle)) \tag{5.8}$$

$$O \vdash Fusion(puddle, T_0). \tag{5.9}$$

In contrast, suppose we have a HOL ontology O' and adopt the definition of $Heat$ in (5.5), but make L polymorphic and assign its type to $\alpha \Rightarrow \beta \Rightarrow \mathbb{R}$ instead, where α and β are type variables – the first argument is the object under consideration and the second argument is the phase change event undergone by the object. To determine the corresponding specific latent heat, the information about phase changes can now be captured in the type without being axiomatised:

$$\forall o,e.\ L(o,e) := \begin{cases} 334 & \text{if } SubstanceOf(o){:}Water \wedge c{:}FusionEvent \\ 2260 & \text{if } SubstanceOf(o){:}Water \wedge c{:}VaporisationEvent \\ \ldots \end{cases} \tag{5.10}$$

where $Water$ is the type of water and $FusionEvent$ and $VaporisationEvent$ are, respectively, the types of fusion and vaporisation events. Rather than having predicates in the language for checking the kind of substance and the phase change, as in (5.6) and (5.7), the specific latent heats in O' depend on the type assigned. This is a more natural adaptation of the underlying Physics, because the kinds of substances and the kinds of phase changes better correspond to sets containing individuals and, thus, logical types. Thus, if the symbol H_2O represents some water and $fusion$ the fusion process H_2O undergoes at time T_0, instead of (5.8) and (5.9), we can now have

$$\{H_2O{:}Water, fusion{:}FusionEvent\} \subset Sig(\mathcal{O}') \tag{5.11}$$

which assigns the type of w to the one representing water undergoing fusion. With a polymorphic representation, we can construct ontologies with fewer explicit axioms and more concise representations. Note that $fusion$ captures the duration of the event.

5.5.4 Modular Representation

5.5.4.1 Issues with Ontological Fault Diagnosis

Most work in the area of ontological fault diagnosis attempts to detect a fault from a single ontology. Even though some work with networked ontologies (Ji et al., 2009), the network is actually collapsible into a single, large ontology. Our method for determining whether the given ontologies require repair focuses on the identification of the pattern describing an underlying fault across multiple ontologies, which is the purpose of the trigger formulae in ORPs. An ontology satisfies some pattern defining a type of fault when an instance of the pattern is derivable from it; in such case, an appropriate repair is brought to bear. In order to meaningfully produce a derivation, the ontology is assumed to be consistent. This is particularly important because all sentences are derivable in inconsistent ontologies, so even irrelevant fault patterns would otherwise be derivable. This would give rise to undesirable or adverse outcomes, as ORPs would be wrongly triggered, resulting in false-positives. Since internal consistency in each ontology in question is assumed, a logical fault, or over-specification, occurs only when a combination is attempted with another ontology that is also internally consistent, inducing global inconsistency (Definition 9, p.34). Thus, in order to identify the specific type of logical fault from which the ontologies suffer, the working environment must contain at least two ontologies. Logical fault diagnosis is not designed to occur after combining together the input ontologies, as the resulting ontology may be inconsistent [3]. Effectively, a single ontology setup will render our proposed method of logical fault diagnosis useless. This shows that a modular representation of ontologies is essential to precise diagnosis of ontological faults.

[3]Note that some ORPs are designed to repair representational rather than logical conflicts, i.e. under-specified ontologies rather than over-specified ontologies.

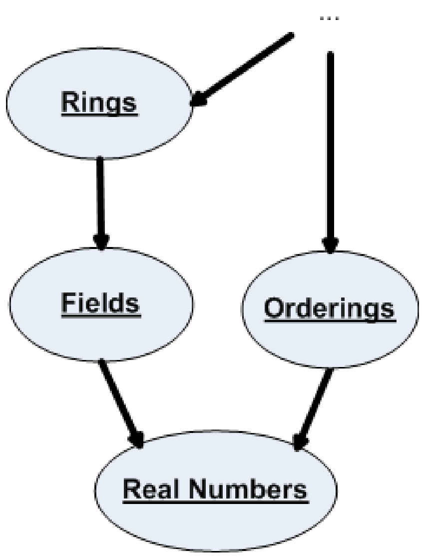


Figure 5.2: A modular representation of the theory of real numbers, where nodes represent ontologies and arcs represent dependencies between ontologies.

5.5.4.2 Issues with Conceptualisation

Knowledge from many domains can be naturally structured using a modular representation. The notion of *extensions* is commonly used for formalising modular structures. If an ontology X is an extension of an ontology Y, then every theorem of Y is a theorem of X and the language of Y is a subset of the language of X. In mathematics, for instance, the theory of real numbers can be viewed as being constructed by extending the theory of fields, which itself may be an extension of the theory of rings. Note that our notion of combination via $\oplus$ (Definition 7, p.34) is, therefore, closely similar to the general notion of extension. Further, because the set of real numbers is ordered, the theory of the reals may also extend the theory of orderings, specifying various abstract orderings. Figure 5.2 illustrates the relationships between the mentioned theories as modular theories, which forms an ontology network[4].

We argue that Physics knowledge also intrinsically exhibits a modular structure, in which Physics theories can be viewed as separate modular theories. One can partition Physics knowledge into modules, which import signature elements and axioms from others and have only an implicit signature declaration and axiomatisation. For instance, classical mechanics contains numerous concepts, e.g., force, energy, and momentum.

[4]More formally, it is a *development graph* (Autexier et al., 2002), which is an acyclic, directed graph where each node represents a theory and links between nodes represent some morphisms allowing one theory to relate to another.

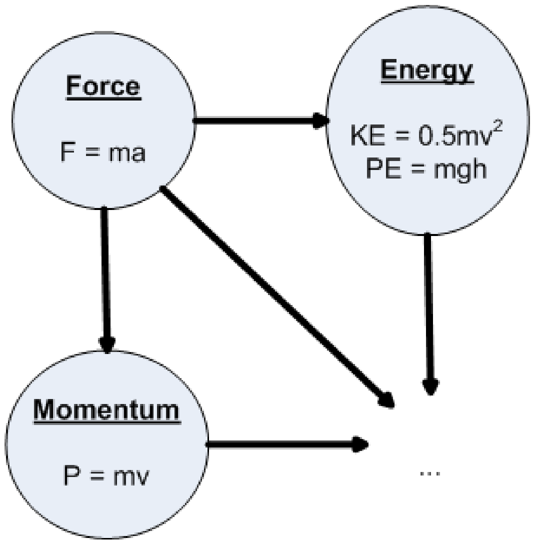


Figure 5.3: A simple modular representation of classical mechanics, where nodes represent ontologies and arcs represent dependencies between ontologies; F, m, a, P, v, KE, PE, g, h denote force, mass, acceleration, momentum, velocity, kinetic energy, potential energy, acceleration due to gravity, and height, respectively.

We can construct a modular representation, in which each major concept belongs to a module containing the relevant equations. The modules can be related to each other by means of extensions, as all these concepts together make up classical mechanics. The resulting structure is typically a directed, *acyclic* graph, as it is not meaningful to have directed cycles in the structure. An example structure is illustrated in Figure 5.3, in which both ontologies *Energy* and *Momentum* extend *Force*, so both of these share the signature and axioms of *Force* but not vice versa, e.g., the signature declarations of m for mass and a for acceleration. If an ontology is to have the same language and logical power as classical mechanics, then it can simply be an extension of all of the three ontologies.

5.5.4.3 Issues with Heterogeneous Ontologies

A goal in the implementation of GALILEO is to ease the restriction on the languages used by modular ontologies, enabling a more flexible approach to modelling (Chan et al., 2010a). It is fairly commonplace in dynamic, unregulated multi-agent environments to deal with ontologies that have different languages; for instance, agents fail to communicate with each other due to a lack of shared understanding of the languages, which is caused by a heterogeneity of languages. If we regard each agent as having its own ontology network, then such heterogeneity arises across multiple discrepant networks. However, heterogeneous ontologies should also be supported within the environment, as supposedly related symbols across different ontologies are sometimes not explicitly related in practice. For instance, in Figure 5.3, the symbol P is declared in the signature of *Momentum*, but not in *Energy*; the symbol KE is in *Energy* but not in *Momentum*. Thus, one may not be able to infer more powerful Physics equations, e.g., $P = \frac{\mathbf{d}(KE)}{\mathbf{d}(v)}$[5] even if we have sufficient abilities to reason over derivatives, from just one of the three ontologies individually, which is not well-formed as P and KE are not both defined in any one of the ontologies. If one is to conjecture about the relationship between P and KE, then there are at least two solutions which both make object-level changes: a) alter the existing dependencies between the ontologies, e.g., make *Momentum* an extension of *Energy* then $P = \frac{\mathbf{d}(KE)}{\mathbf{d}(v)}$ would be a theorem of *Momentum*; or b) create a new ontology that extends both *Momentum* and *Energy*, i.e. $Momentum \oplus Energy$, and assert the relationships between the symbols under consideration.

We prefer b) over a) because modifying the existing dependencies between ontologies results in a more significant violation of the commitment to the model than introducing new ontologies into the model does, which can be seen as merely incorporating new knowledge. Adding a new ontology into a network preserves the theorems in the ontologies of the original setup, but changing the dependencies between ontologies in the described manner introduces new theorems in $Momentum$.

5.5.4.4 Issues with Ontology Repair

Another key benefit of adopting a modular representation of ontologies is a better management of evolved ontologies. Not every ontology in a network requires repair when a fault is detected. Especially in the Physics domain, some ontologies may be protected from repair if they are valued at high confidence, e.g., the ontologies specifying arithmetic and foundational mathematics, or the purpose of the ontologies is merely to aid inference and they are not susceptible to repair, e.g., the ontologies specifying biological knowledge when the working domain is Physics – evolving Biology ontologies would be beyond the scope of interest. It would be extremely troublesome to selectively effect the repair if a flat representation is used, as all axioms of the ontology

[5] Note that this is not a general relationship between kinetic energy and momentum, but only for Newtonian point particles.

are supposed to be equally susceptible to alteration. Meta-data may also be needed to discriminate the susceptible axioms from others. Moreover, each repair attempt would create new whole ontologies, so it would render the extraction of interesting information, such as which ontologies are (not) affected by the repair, impossible.

5.6 Representation and Handling of Heterogeneity

The GALILEO system is designed to handle flexible ontological configurations, such as modular ontologies (§5.5.4) and heterogeneous ontologies, which are ontologies that have distinct languages. It is often impractical to construct ontologies that are restricted to share a common language, especially when modelling complex real-world knowledge. An ontology typically captures a particular view of the world, so the language used to describe such a partial view is inevitably non-exhaustive and incomplete. For instance, based on the representation depicted in Figure 5.3, suppose we have two ontologies *Momentum* and *Energy*:

$$Sig(Momentum) := \{P{:}Obj \Rightarrow Mom \Rightarrow \mathbb{R},\ m{:}Obj \Rightarrow Mom \Rightarrow \mathbb{R}, \tag{5.12}$$

$$v{:}Obj \Rightarrow Mom \Rightarrow \mathbb{R}\} \tag{5.13}$$

$$Ax(Momentum) := \{P(o,t) ::= m(o,t) \times v(o,t)\} \tag{5.14}$$

$$Sig(Energy) := \{KE{:}Obj \Rightarrow Mom \Rightarrow \mathbb{R},\ m{:}Obj \Rightarrow Mom \Rightarrow \mathbb{R}, \tag{5.15}$$

$$v{:}Obj \Rightarrow Mom \Rightarrow \mathbb{R}, \dots\} \tag{5.16}$$

$$Ax(Energy) := \{KE(o,t) = \frac{1}{2} m(o,t) \times v(o,t)^2, \dots\} \tag{5.17}$$

Representing $P(o,t) = \frac{\mathbf{d}KE}{\mathbf{d}v}$ in either *Momentum* or *Energy* is not well-formed. So we need to introduce an ontology that is expressive enough to represent relationships between the symbols under consideration.

The ability to perform inference may be limited due to the problem of heterogeneity as well. Disagreements between the languages of ontologies may only be syntactic, so their relationships depend on the interpretation of the ontologies. For instance, suppose we have two ontologies O and O' and assume same symbols represent same things, i.e.

alignments are not needed:

$$Sig(O) := \{a{:}\alpha,\ b{:}\beta,\ f{:}\alpha \Rightarrow \beta \Rightarrow Nat, \quad g{:}\alpha \Rightarrow \beta \Rightarrow Nat,\ h{:}\alpha \Rightarrow \beta \Rightarrow Nat\} \tag{5.18}$$

$$Ax(O) := \{f(x,y) := g(x,y) \times h(x,y),\ h(a,b) := 1\} \tag{5.19}$$

$$Sig(O') := \{a{:}\alpha,\ b{:}\beta,\ m{:}\alpha \Rightarrow \beta \Rightarrow Nat\} \tag{5.20}$$

$$Ax(O') := \{m(a,b) := 2\} \tag{5.21}$$

where the languages of O and O' syntactically share some common symbols, e.g., a and b, but there are also differences between them as m is not in the signature of O, whereas f, g, and h are not in the signature of O'. Even with O and O' merged, the inference of $f(a,b)$ cannot proceed beyond $f(a,b) = g(a,b)$. Since ontological representations are merely syntactic structures, depending on the interpretation of O and O', the seemingly distinct symbols might be (physically) related and allow to further the inference of $f(a,b)$. For instance, $f(x,y)$ in O might represent the distance an object x has travelled over an event y, $g(x,y)$ might represent the average velocity of x during the event y, and $h(x,y)$ might represent the duration of x being in event y. If $m(x,y)$ in O' represents the empirical measurement of the distance of x over y using an odometer, then we can directly relate m in O' to g in O. Alternatively, m might instead be related to f and h in O', e.g., in the form of $m(x,y) = \frac{f(x,y)}{h(x,y)}$, which provides the same amount of logical power as the direct relationship. With either of these relationships, $f(a,b) = 2$ can then be inferred, with the interpretation that the distance object a has travelled over event b being 2.

The need for handling heterogeneity in language is particularly appropriate to Physics, as Physics theories are commonly contrasted with empirical evidences. Theories often do not speak in the language of experiments, but in one about the relevant physical concepts, e.g., force, mass, and energy. Experiments, on the other hand, are described by the language of the apparatus or other empirical instruments, e.g., thermometer, spectral graph, and odometer. Thus, in order to make sense of the consequences from merging a theory with empirical data, we require a mechanism for addressing the heterogeneity between seemingly discrepant languages – which we call *bridging axioms*. Note that these can also be used for alignment, e.g., $f = g$ when f and g are two different representations of the same concept.

5.6.1 Bridging Axioms

In the object-level of the representation, the languages of even originally unrelated ontologies can be related by creating a new ontology that is the result of combining them. If an ontology O_e is the result of combining together multiple ontologies O_a, O_b, ... with signatures $Sig(O_a)$, $Sig(O_b)$, ..., respectively, then O_e has the capacity to use the symbols imported from all O_a, O_b, ... in its local axioms, since $Sig(O_a) \cup Sig(O_b) \cup \cdots \subseteq Sig(O_e)$. As the relationships between symbols in $Sig(O_a)$, $Sig(O_b)$, ... are expressed in the local axioms of O_e, the additional logical strength can only be leveraged if O_e, or other ontologies that combine with O_e, are used in the inference. Being the result of a combination also means that every theorem of the original ontologies, O_a, O_b, ..., is a theorem of it. Thus, if the reasoning task requires an agreement between the originally symbols about which originally there was a disagreement, then O_e can become the focus for reasoning.

The relationships between the symbols under consideration are specified in *bridging axioms*, which are axioms that relate together the truths across ontologies. The encoding of bridging axioms is based on McCarthy's *lifting axioms* (McCarthy and Buvac, 1998), which is defined as

> ...axioms which relate the truth in one context to the truth in another context.

Lifting axioms are generally in the following form:

$$ist(\kappa_1, \phi_1) \wedge \cdots \wedge ist(\kappa_n, \phi_n) \supset ist(\kappa, \phi) \tag{5.22}$$

where $ist(\kappa, \phi)$ means that the formula ϕ is true in a context κ[6]. The lifting axiom above means that the formula ϕ is true in a context κ if ϕ_i for $1 \leq i \leq n$ is true in $\mathcal{K}_i$ for $1 \leq i \leq n$, respectively. Lifting axioms themselves are encoded in a context that is expressive enough to represent all formulae in all contexts. For our application, we encode bridging axioms in an ontology that extends the ontologies under consideration by combining them all via the $\oplus$ operator. We call the meta-ontology containing a bridging axiom a *bridging ontology*. A bridging ontology is not an object-level ontology but an ontology at the meta-level. Bridging axioms relate the provability of sentences in one ontology to the provability of sentences in another. If we treat contexts as ontologies, we can formulate (5.22) as a bridging axiom:

$$\kappa_1 \vdash \phi_1 \wedge \cdots \wedge \kappa_n \vdash \phi_n \longrightarrow \kappa \vdash \phi \tag{5.23}$$

[6]Note that $\supset$ can be replaced by $\longrightarrow$ without changing meanings.

where $\kappa \vdash \phi$ means that ϕ is a theorem of κ.

For representing that P is the derivative of KE w.r.t v (5.12 - 5.17), bridging axioms (premises) can be used to derive the following:

$$(\forall a.\ (Momentum \vdash m = a \longleftrightarrow Energy \vdash m = a) \wedge$$
$$\forall a.\ (Momentum \vdash v = a \longleftrightarrow Energy \vdash v = a)) \longrightarrow$$
$$Momentum \oplus Energy \vdash P = \frac{\mathbf{d}(KE)}{\mathbf{d}(v)}$$

where $Momentum \vdash P = m$ means that $P = m$ is a sentence derivable in the ontology *Momentum*. The new ontology, $Momentum \oplus Energy$, imports the signatures and all theorems from *Momentum* and *Energy*, so it is expressive enough to represent $P = \frac{\mathbf{d}(KE)}{\mathbf{d}(v)}$ within it, which is a theorem provided that we have support for reasoning about derivatives.

For inferring $f(a,b) = 2$ (5.18 - 5.21), bridging axioms can be used to derive the following:

$$(\forall a.\ (O \vdash g(x,y) = a \longleftrightarrow O' \vdash m(x,y) = a) \longrightarrow$$
$$O \oplus O' \vdash m = g$$

Bridging axioms resemble lifting axioms in various ways. For instance, contexts in lifting axioms are essentially ontologies and the *ist* predicate can be associated with $\vdash$, as a formula provable in some ontology if and only if it is a consequence of premises assumed to be true.

5.6.2 Factorisation

In order to design a highly general mechanism for detecting faults in multiple ontologies, we must enable at least some inference to be performed across even globally inconsistent ontologies. For instance, given a predictive ontology containing definitions and a sensory ontology containing pure empirical data, a conflict is deducible only if the definitions are instantiated by the empirical data. Clearly, reasoning with the combination of these globally inconsistent ontologies is virtually meaningless, but the inconsistency itself may be circumvented by excluding certain axioms from the combination. Our solution is to enable ORPs to be performed not on the original input ontologies themselves, but on the ontologies in a *factorised representation* of the original.

Definition 11 (Factorised Representation) *The* factorised representation $\mathcal{F}(O)$ *of an ontology* O:$\mathbb{O}$ *is defined as:*

$$\mathcal{F}(O) ::= \{O' \mid O' \preceq O\}$$

where

$$\langle S', A' \rangle \preceq \langle S, A \rangle \Longleftrightarrow S' = S \wedge A' \subseteq A \wedge A' \neq \emptyset.$$

A factorised representation of an ontology with n axioms is essentially a set of ontologies in which each ontology contains a unique k-combination of original axioms for $1 \leq k \leq n$, The idea of working with a factorised representation of ontologies is loosely based on the concept of a factorised representation of a set of independent variables in Bayesian Networks (Jensen, 1996). Consequently, each ontology in a factorised representation is a sub-ontology of the ontology being factorised. The models satisfying any subset of the original axioms must also satisfy an ontology in the factorised representation and the models satisfying any ontology in the factorised representation must also satisfy a subset of the original axioms. Subontologies have been proven useful and adopted to solve specific reasoning tasks about ontologies (Haase et al., 2005; Du et al., 2008). We will use the subscript $_f$ and multiples thereof to denote the ontologies in a factorised representation, e.g., O_f, i.e. is a sub-ontology of the original ontology, O.

Proposition 1 *Given an ontology* O:$\mathbb{O}$*, the following holds:*

$$\forall A_f \subseteq Ax(O).\ \exists O_f \in \mathcal{F}(O).\ Ax(O_f) = A_f.$$

Proposition 2 *Given an ontology* O:$\mathbb{O}$*, the following holds:*

$$\forall O_f \in \mathcal{F}(O).\ \exists A_f \subseteq Ax(O).\ Ax(O_f) = A_f.$$

From Propositions 1 and 2, we claim that the standard interpretation of an ontology, which itself should be a standard interpretation of at least one of the original axioms, is a standard interpretation of at least one of the ontologies in the corresponding factorised representation. This is because the factorised representation must contain all possible sub-ontologies of the corresponding ontology. Working with a factorised representation, therefore, essentially enables reasoning to be performed over any subset of the axioms of the input ontologies. In the case where the environment contains multiple factorised representations, each corresponding to a different input ontology, an ontology in one factorised representation could be combined with one in another. In effect, a subset of the axioms of an input ontology can now be combined with the whole of another input ontology. This is a powerful approach and increases the generality of the repair mechanism, because it enables some reasoning across globally inconsistent ontologies – the reasoning can be focussed on only those ontologies in the factorised representations that do *not* contain the axioms that would otherwise induce global inconsistency. The combination operator $\oplus$ enables the reuse of ontologies for building factorised representations as networks, so that the number of ontologies created can be kept to a minimum.

We have provided a Python script for producing a factorised representation of an input ontology (B.2)

5.7 Summary

This chapter introduces the GALILEO system, which is discussed in more detail in §6 and §7. The implementation of our ideas is tailored to an exact context for the theory to be used: ontology repair in Physics. In addition, the production of such a system has forced us to make many decisions as to how to contain the system sufficiently so that a fully working version can be produced within the time scale. This has led to the production of a system that is not as comprehensive as we would like, which is inevitable when tackling such a vast problem from a novel direction.

The implementation of GALILEO contains about 7,000 lines of ML code, which interfaces with Isabelle, and 50 lines of Python code for producing factorised representations (§5.6.2). The system we have produced is capable of:

1. Starting with an Isabelle theory file containing the input ontologies.
2. Processing interactive input from the user to invoke and minimally guide an ORP.
3. Detecting ontological faults and producing logically valid diagnoses.
4. Filtering out physically meaningless diagnoses.
5. Performing repair to the input ontologies depending on the selected diagnosis.
6. Relating newly repaired ontologies to the original.

We claim that points 2, 3, 4, and 5 represent original research, whilst the rest of the functionality of the system is necessary to provide a platform from which to implement this original research. Points 3, 4, and 5 represent the central focus of the project.

Chapter 6

Mechanising Conflict Diagnosis

6.1 Introduction

The first part of executing an ORP is to verify that the trigger follows from the given ontologies. GALILEO ensures that the fault which the ORP is designed to resolve indeed exists within the input ontologies, preventing giving false positive results. To this end, we encode ontologies as locales. The system generates a proof obligation that is formulated in a way such that reasoning across multiple ontologies can be achieved. The proof obligation is required to be discharged by the user in order to proceed to repair. For certain ORPs, the system discovers physically meaningful terms that are potential candidates for being the culprit in causing the detected conflict. Note that, by an ontology fault or conflict, we mean one of over-specification or under-specification (§9). Because most of the ORPs involve dealing with the application of one polymorphic variable to another, higher-order unification expectedly returns an unmanageably huge number of unifiers. We, therefore, incorporate heuristics for pruning the search space and eliminating physically meaningless results.

6.2 Ontological Fault Diagnosis with Isabelle

Conflict diagnosis is the task of discovering the instantiations of *stuff* and other variables, so that ORPs can be properly triggered; GALILEO stores these instantiations as they are needed to repair the detected conflict. In a typical environment for GALILEO to perform ontology evolution, higher-order reasoning is involved at both the object- and meta-levels in the ORPs. The object-level is used to represent the ontologies, which are essentially a collection of physics equations asserted as axioms; the meta-level to represent the ORPs, e.g., for diagnosing a conflict in the ontologies and manipulating the ontologies by applying repair operations to them[1]. There are, therefore, two separate domains of quantification: the object-level reasons about physical entities, concepts, observations, and so forth; the meta-level about object-level ontologies, formulae, types, and so forth, e.g., it reasons about whether some object-level ontology has a theorem matching some syntactic pattern and then adds or retracts some axioms or signature symbols. This means there could potentially be two logics to encode: one for the object-level and one for the meta-level. In this approach, the object-logic can be deeply embedded within the meta-logic by providing an explicit higher-order abstract syntax for the ontologies. The meta-logic can then perform matching with a syntactic pattern by recursing over the abstract syntactic structures of object-level formulae. However, if we want to do higher-order inference at the object-level, then higher-order unification needs to be implemented from scratch, which is not desirable[2]. Fortunately, some systems, including Isabelle and Twelf (Pfenning and Schürmann, 1999), are built on a logical framework and object logics are encoded via the meta-logic, so it inherently distinguishes between object- and meta-logics and higher-order unification is implemented as a meta-level procedure. It is, therefore, appropriate to employ a reasoner that offers us to encode physical ontologies in the object-logic and specify the ORPs in the meta-logic.

Since we work with HOL ontologies with polymorphic types, an appropriate reasoner must provide an environment that supports reasoning in such a logic and type theory. The validity problem in HOL is undecidable in general[3], so higher-order theorem provers such as Isabelle (Paulson, 1994), TPS (Andrews et al., 1984), Nuprl (Constable et al., 1986), Coq (Bertot and Castéran, 2004) and others typically require user interaction to produce significant proofs. Thus, these theorem provers are often perceived as *proof assistants* rather than *automated theorem provers*, because a human user is typically expected to guide the search for proofs, whilst the system provides feedback for each proof step. That said, many of these theorem provers support some form of proof automation, e.g., applying tactics, generalised equational rewriting, and even translating a given problem into first-order logic and solving it with first-order provers. When the automation fails, the system defaults to user interaction.

[1]The ORPs are phrased in Isabelle/ML.
[2]This would likely to be the case had we continued to adopt λ-Prolog for our implementation.
[3]The validity problem is undecidable even in pure first-order logic.

We have chosen Isabelle not only because it comes with an implementation of higher-order unification, but that it also has a vast library of already formalised mathematical theories, among which are integers and real numbers, as well as sets and vectors, which are all particularly useful for reasoning about physics problems. Isabelle already has several logics, e.g., Isabelle/HOL, formalised in its meta-logic, Isabelle/Pure, its meta-logic, so we can naturally encode the ontologies in the object-logic and perform any required inference. Isabelle, being an interactive theorem prover, provides a robust interactive environment for collaborating with the user during the process of proof search, which is useful for capturing a user's input during the ontology evolution process, when required. Overall, we consider Isabelle's framework to be adequate for providing reasoning support for the kind of high expressivity in our logic and a good environment for evolving ontologies using Isabelle's implementation of modular reasoning.

6.2.1 Ontologies as Contexts

As already discussed, an ontology is a specification of a conceptualisation, which includes the objects, concepts, and other possible entities in the domain of discourse. Furthermore, we regard an ontology to be a logical theory with a signature and a set of axioms. A specification of an ontology may represent a general conceptualisation about the world, which is an abstraction that encapsulates an aggregation of multiple situations. For instance, the physics equation $\forall o{:}Obj, t{:}Mom.\ TE(o,t) ::= KE(o,t) + PE(o,t)$ quantifies over all objects and time moments in the domain. Multiple instances of the same equation may coexist in different ontologies, so an appropriate machinery for representation must allow instances of axioms to be created. Further, our application requires reasoning across multiple ontologies; for instance, we need to verify whether some function measuring a certain property returns the same value in two different ontologies.

Modelling ontologies as contexts enables the encoding of a view of a domain to be localised. As such, reasoning across multiple ontologies can, therefore, be realised. Modularising the encodings of views into contexts helps us analyse the shape of ontological conflicts between them. If several contexts are in conflict with each other, we can isolate each context individually and apply the repair that is relevant to the corresponding encoding.

Typically, each signature symbol of an Isabelle locale corresponds to some entity in the domain of discourse, so all entities encompassed in the conceptualisation are declared in the signature. The axioms must be specified in the corresponding language in order to be well-formed. This effectively implements *parametric theories* where the axioms are specified w.r.t. the values of the parameters. So, the use of Isabelle's locales lets us create instances of axioms by giving values to the parameters and, thus, the axioms of a locale. Modelling ontologies as locales means that we specifically view ontologies as parametric theories. This conveniently allows different instantiations of a specification of an ontology to be produced by using different parameter values. Such a configuration means that the concepts in each instantiation correspond to different interpretations of the same concepts – similar to the different interpretations of a concept, e.g., letter A, after teaching it to a class of students. Formalising ontologies as contexts, and thus parametric theories, is clearly powerful for performing reasoning in multiple ontology environments.

The problem GALILEO solves involves detecting faults across multiple ontologies, so each ontology is formulated as a locale with a set of parameters and axioms, and possible dependencies on other locales within the global context.

6.2.2 Variable Sharing

All ORPs are designed to have the variables, e.g., *stuff*, instantiated during a process of inference of inferring the corresponding trigger formula. When the variable *stuff* appears in the syntactic patterns that are used to match against theorems provable in different ontologies, it is required to be instantiated to a value shared by the ontologies under consideration. Merely combining the conflicting ontologies together and inferring the trigger formula in the resulting ontology would often induce inconsistency. The trigger formula can then always be inferred, the inference itself is trivial, since all formulae are theorems of inconsistent ontologies.

There are potentially two alternatives to handling variable sharing: a) reasoning about the trigger formula within a locale that extends from the locales corresponding to the ontologies under consideration, and b) directly formulating the reasoning problem in a meta-level mechanism.

Alternative a) is plausible because each locale is mostly made up of a set of parameters and a set of axioms, and each locale can be independently instantiated based on the provided values to the parameters. Reasoning across these locales is a matter of formulating the problem in terms of the values that instantiate the parameters. A shortcoming of this approach is that, since the locales being extended are instantiated using different values to avoid inconsistency, *stuff* will not be instantiated to a common value without some meta-level method for helping the instantiation. The customised method of instantiation will most likely involve binding together the related values used to instantiate the various locales, which is a rather unnatural procedure. Another shortcoming is that a new locale needs to be created in the environment each time we want to reason across multiple ontologies. The ontology created does not naturally correspond to a physical ontology and is merely used to facilitate the reasoning process. Furthermore, it does not accurately emulate the formalisations of the ORPs, where a conflict is expressed as a combination of sentences that are theorems of the individual ontologies. Some meta-level mechanism is still required to generate the proof goal corresponding to the relevant ORP and ensure that *stuff* is instantiated to the correct parameter value.

A more desirable option is, therefore, to adopt b) and implement a meta-level mechanism that generates an object-level proof goal based on the given parameters of the locales under consideration without creating helper locales. If the locales provided are `M te1 ball1 t1` and `N te2 ball2 t2`, where `te1`, `ball1` and `t1` are the values of the parameters of `M` and `te2`, `ball2` and `t2` are the parameters of `N`, then the proof goal is formulated directly in terms of these values without creating new locales. Suppose `M` and `N` share the same signature and we postulate that `te1(ball1,t1)` and `te2(ball1,t1)` have different values and try to infer the WMS trigger formula in them, then the proof goal for this task is:

$$\exists v.\ (\forall te1, ball1, t1.\ M(te1, ball1, t1) \Longrightarrow te1(ball1, t1) \geq v) \wedge \\ (\forall te2, ball2, t2.\ N(te2, ball2, t2) \Longrightarrow te2(ball2, t2) < v) \wedge \ldots$$

where M and N are ternary predicates. If the goal can be discharged, then *stuff* here is instantiated to a subexpression of $te1(ball1, t1)$, because we already know M and N have the same signature and both $te1$ and $te2$ must refer to the same symbol – likewise for other values. The user can utilise all reasoning facilities that are provided by Isabelle to search for a proof. We have adopted this approach in GALILEO and the generated proof goal for each ORP is summarised below.

6.2.2.1 Proof Obligation for Triggering *Where's My Stuff*

Suppose

- $M{:}\alpha_1 \Rightarrow \alpha_2 \Rightarrow \cdots \Rightarrow \alpha_m \Rightarrow bool$ is the locale representing the ontology whose value of $f(\mathit{stuff})$ is greater, i.e. the instantiation of O_1 in Figure 4.2, p.49 and N:$\beta_1 \Rightarrow \beta_2 \Rightarrow \cdots \Rightarrow \beta_n \Rightarrow \mathit{bool}$ is the locale whose value of $f(\mathit{stuff})$ is smaller, i.e. the instantiation of O_2;

- *fst* is instantiated to the term being postulated to create a conflict, which is postulated to return different values in different ontologies, i.e. the instantiation of $f(\mathit{stuff})$;
- p_i for $1 \le i \le m$ are the parameters of M, where M has m parameters;
- q_i for $1 \le i \le n$ are the parameters of N, where N has n parameters.

When WMS is invoked, GALILEO automatically generates the following proof obligation:

$$\exists v.\ (M(p_1,p_2,\ldots,p_m) \Longrightarrow \mathit{fst} > v) \wedge (N(q_1,q_2,\ldots,q_n) \Longrightarrow \mathit{fst} \le v) \vee \\ (M(p_1,p_2,\ldots,p_m) \Longrightarrow \mathit{fst} \ge v) \wedge (N(q_1,q_2,\ldots,q_n) \Longrightarrow \mathit{fst} < v) \quad (6.1)$$

where p_i for $1 \le i \le m$ and q_i for $1 \le i \le n$ are required to be labelled with the same name if they refer to the same parameter and with the same name as used when the locales for the ontologies are originally defined. For instance, suppose M and N share a common signature containing a function symbol *TE*, then the antecedents in (6.1) can be $M(te)$ and $N(te)$ or other variants with te replaced by other names. Internally, GALILEO renames the values supplied in order to instantiate the locales without inducing inconsistency[4]. The proof goal has the same shape as the WMS trigger formula (4.13 - 4.14, 49), except only v is explicitly quantified, because the user instantiates the conflicting ontologies and the conflicting term, $f(\mathit{stuff})$ – thus, implicitly their types as well. Commands for invoking WMS in an example execution are depicted in Figure 6.1. The command `wms` invokes the WMS repair plan; `try_o1` and `try_o2` take the locales representing the ontologies under consideration, i.e. O_1 and O_2, respectively; `try_fstuff` takes the instantiation of $f(\mathit{stuff})$; and, `verify` requests the generation of the proof goal in the shape of (6.1). Although the user guides the evolution process by indicating the instantiation of $f(\mathit{stuff})$, the system still needs to identify the actual instantiation of *stuff* in order to perform the repair.

6.2.2.2 Proof Obligation for Triggering Inconstancy

Suppose

[4]Typically, a number is appended to each value.

```
wms
try_o1 "Os1 endev drp vel te pe ke mass g ball sysball"
try_o2 "Os2 endev drp vel te pe ke mass g posn photoat ball sysball"
try_fstuff "te sysball (endev drp)"
verify
```

Figure 6.1: Example commands for verifying that the given ontologies contain a WMS-type of fault, where 'Os1' and 'Os2' are two different locales sharing most signature symbols.

- $M{:}\alpha_1 \Rightarrow \alpha_2 \Rightarrow \cdots \Rightarrow \alpha_m \Rightarrow bool$ is the locale representing the theoretical ontology, which contains the original definition of *stuff*, i.e. the instantiation of O_x (Figure 4.5, 60)
- $N^1{:}\beta_1 \Rightarrow \beta_2 \Rightarrow \cdots \Rightarrow \beta_{n_1} \Rightarrow bool$ and $N^2{:}\beta_1 \Rightarrow \beta_2 \Rightarrow \cdots \Rightarrow \beta_{n_2} \Rightarrow bool$ are the locales representing the observational ontologies
- *fst* is the term being postulated to create a conflict, i.e. the instantiation of $f(stuff)$
- p_i for $1 \leq i \leq m$ are the parameters of M
- q_i^1 for $1 \leq i \leq n^1$ are the parameters of N^1 and q_i^2 for $1 \leq i \leq n^2$ are the parameters of N^2
- $cond^1$ is the condition under which the observation in N^1 is made
- $cond^2$ is the condition under which the observation in N^2 is made

When Inconstancy is invoked, GALILEO automatically generates two proof goals. To check that the given locales contain a fault similar to that targeted by Inconstancy, the first goal to be discharged is:

$$\begin{aligned} \exists c_1, c_2, v_1, v_2.\ (M(p_1, p_2, \ldots, p_m) \Longrightarrow c_1 \neq c_2 \wedge v_1 \neq v_2) \wedge \\ (N^1(q_1^1, q_2^1, \ldots, q_{n^1}^1) \Longrightarrow fst = c_1 \wedge cond^1 = v_1) \wedge \\ (N^2(q_1^2, q_2^2, \ldots, q_{n^2}^2) \Longrightarrow fst = c_2 \wedge cond^2 = v_2) \end{aligned} \tag{6.2}$$

where all parameters are required to be labelled with the same name if they refer to the same parameter. The proof goal has the same shape as the Inconstancy trigger

```
inconstancy
try_o1 "Os1 g"
try_o2 "Os2 star1 g accl" "accl star1"
try_o3 "Os3 star2 g accl" "accl star2"
try_fstuff "g"
verify
```

Figure 6.2: Example commands for verifying that the given ontologies contain an Inconstancy-type of faultwhere ‘Os1’, ‘Os2’ and ‘Os3’ are three different locales.

formula (4.46 - 4.48, p.60), except that the original definition of *stuff* is not proved to be derivable from the theoretical ontology here, which is done in a separate step (§6.4). The reason is that we do not know how *stuff* is instantiated at this point. So, this is only part of the diagnosis process for Inconstancy; the diagnosis continues once the instantiation of *stuff* is chosen. Only c_1, c_2, v_1, and v_2 are explicitly quantified here, as the user supplies the ground instantiations of O_x, $O_{y,1}(v(b_1) =_{\tau'''} v_1)$, $O_{y,2}(v(b_2) =_{\tau'''} v_2)$, O_z, and $f(stuff)$. Commands for invoking Inconstancy in an example execution are depicted in Figure 6.2. The command `inconstancy` invokes the Inconstancy repair plan; `try_o1` takes the locale representing the theoretical ontology; `try_o2 X Y` and `try_o3 X Y` capture the locales representing the ontologies (X) and their respective condition vectors (Y); `try_fstuff` takes the instantiation of $f(stuff)$; and, `verify` requests the generation of the proof goal (6.2). Similar to WMS, although the user guides the evolution process by indicating the instantiation of $f(stuff)$, the system still needs to identify the specific instantiation of *stuff* in order to perform the repair.

6.2.2.3 Proof Obligation for Triggering Reidealisation

Suppose

- $M{:}\alpha_1 \Rightarrow \alpha_2 \Rightarrow \cdots \Rightarrow \alpha_m \Rightarrow bool$ and $N{:}\beta_1 \Rightarrow \beta_2 \Rightarrow \cdots \Rightarrow \beta_n \Rightarrow bool$ are the instantiations of O_1 and O_2 in Figure 4.3, p.54, respectively
- *fst* is the term being postulated to create a conflict, i.e. the instantiation of $f(stuff)$
- p_i for $1 \le i \le m$ are the parameters of M
- q_i for $1 \le i \le n$ are the parameters of N

```
reidealisation
try_o1 "Os1 pluto kuiperbelt"
try_o2 "Os2 pluto plutinoa kuiperbelt coorda"
try_fstuff "ClearNeighbour pluto" "True"
try_type "DPlanet"
verify
```

Figure 6.3: Example commands for verifying that the given ontologies contain a Reidealisation-type of fault, where ‘Os1’ and ‘Os2’ are three different locales and ‘DPlanet’ is a type.

When Reidealisation is invoked, GALILEO automatically generates the following proof obligation:

$$\exists v.\ (M(p_1, p_2, \ldots, p_m) \Longrightarrow fst \neq v) \wedge (N(q_1, q_2, \ldots, q_n) \Longrightarrow fst = v) \tag{6.3}$$

where, similar to the previous repair plans, p_i for $1 \leq i \leq m$ and q_i for $1 \leq i \leq n$ are required to be labelled with the same name if they refer to the same parameter. The proof goal has the same shape as the Reidealisation trigger formula (4.31 - 4.33, p.54), except that only v is explicitly quantified, because the instantiations of the ontology variables and $f(stuff)$ are supplied. Further, the value of $f(stuff)$ if *stuff* was to take the type τ_2 is not yet checked – it is done after the instantiation of *stuff* is identified. Commands for invoking Reidealisation in an example execution are depicted in Figure 6.3. The command `reidealisation` invokes the Reidealisation repair plan; `try_o1` and `o_2` take the locales representing the theoretical and sensory ontologies; `try_fstuff` takes the instantiation of $f(stuff)$; `try_type` takes the instantiation of τ_2 as an argument; and, `verify` requests the generation of the proof goal in the shape of (6.3). Similar to WMS and Inconstancy, although the user guides the evolution process by indicating the instantiation of $f(stuff)$, the system needs to discover the specific instantiation of *stuff* in order to perform the repair.

6.2.2.4 Proof Obligation for Triggering Unite

Suppose

- M:$\alpha_1 \Rightarrow \alpha_2 \Rightarrow \cdots \Rightarrow \alpha_m \Rightarrow bool$ is the instantiation of O (Figure 4.6, p.64) *stuff*

- $MExt_1$:$\alpha_1 \Rightarrow \alpha_2 \Rightarrow \cdots \Rightarrow \alpha_m \Rightarrow bool$ is a consistent extension of M in which the equality between $stuff_1$ and $stuff_2$ is provable

- $MExt_2$:$\alpha_1 \Rightarrow \alpha_2 \Rightarrow \cdots \Rightarrow \alpha_m \Rightarrow bool$ is a consistent extension of M in which the inequality between $stuff_1$ and $stuff_2$ is provable

- fst_1 and fst_2 are the instantiations of $f(stuff_1)$ and $f(stuff_2)$, respectively

- p_i^1 for $1 \leq i \leq m$ are the parameters of $MExt_1$

- p_i^2 for $1 \leq i \leq m$ are the parameters of $MExt_2$

When Unite is invoked, GALILEO automatically generates the following proof obligation:

$$(MExt_1(p_1^1, p_2^1, \ldots, p_m^1) \Longrightarrow fst_1 = fst_2) \wedge \\ (MExt_2(p_1^2, p_2^2, \ldots, p_m^2) \Longrightarrow fst_1 \neq fst_2) \tag{6.4}$$

where, similar to the previous repair plans, all parameters are recommended to be normalised by the user so that parameters that bind to the same symbol are indicated by the same label. The proof goal here only verifies the unprovability of $fst_1 = fst_2$ in O_1 by checking whether $fst_1 = fst_2$ is provable in one consistent extension and $fst_1 \neq fst_2$ is provable in another consistent extension. We do not know how *stuff* is instantiated at this point, so we cannot proceed to check that the two stuffs have the same defining property. So, this is only a part of the diagnosis process for Unite; once the instantiation of *stuff* is chosen, the diagnosis continues (§6.4). Commands for invoking Unite in an example execution are depicted in Figure 6.4. The command `unite` invokes the Unite repair plan; `try_o1` and `o_2` take the locales representing the theoretical and sensory ontologies as input; `try_o1ext1` and `try_o1ext2` each takes an extension of the input locale to `try_o1` as input; `try_defprop` takes the instantiations of *dp* and τ in (4.64, p.64) as arguments; `try_fstuff1` and `try_fstuff2` take the instantiations of $f(stuff_1)$ and $f(stuff_2)$, respectively; and, `verify` requests the generation of the proof goal (6.4).

6.2.2.5 Proof Obligation for Triggering Spectrum

Suppose

```
unite
try_o1 "Os1 morningstar simsighting"
try_o2 "Os2 morningstar eveningstar orbit window position"
try_o1ext1 "Os1Ext1 morningstar eveningstar simsighting"
try_o1ext2 "Os1Ext2 morningstar eveningstar simsighting"
try_defprop "orbit" "CelestialObj"
try_fstuff1 "morningstar"
try_fstuff2 "eveningstar"
verify
```

Figure 6.4: Example commands for verifying that the given ontologies contain a Unite-type of fault, where 'Os1' and 'Os2' are two different locales; 'Os1Ext1' and 'Os1Ext1' are two different extensions of 'Os1'.

- $M{:}\alpha_1 \Rightarrow \alpha_2 \Rightarrow \cdots \Rightarrow \alpha_m \Rightarrow bool$ is the locale in which neither the equality nor the inequality between the two *stuff*s can be proved, i.e. the instantiation of O_1 (Figure 4.7, p.68)
- $Objs$ is the instantiations of Q
- $Preds$ is the instantiations of P

When Spectrum is invoked, GALILEO automatically generates the following proof obligation:

$$M(p_1, p_2, \ldots, p_m) \Longrightarrow card(Objs) > 1 \wedge card(Preds) > 1 \wedge \quad \forall o \in Objs. \exists! p \in Preds. p(o) \tag{6.5}$$

where $card(S)$ returns the cardinality of the set S and, similar to all previous repair plans, are required to be labelled with the same name if they refer to the same parameter. The proof goal has the exact same shape as the Spectrum trigger formula (4.75, p.68). Commands for invoking Spectrum in an example execution are depicted in Figure 6.5. The command `spectrum` invokes the Spectrum repair plan; `try_o` takes the locale representing the ontology under consideration as input; `try_p` takes the set containing all unary predicates as input; and `try_p` takes the set containing all objects as input; `verify` requests the generation of the proof goal (6.5).

```
spectrum
try_o "Os1 isred isblue isgreen ball1 ball2 ball3 ball4 obs relatedpreds"
try_p "relatedpreds"
try_q "obs"
verify
```

Figure 6.5: Example commands for verifying that the given ontologies contain a Spectrum-type of fault, where ‘Os1’ is a locale.

```
wms
try_o1 "Os1 endev drp vel te pe ke mass g ball sysball"
try_o2 "Os2 endev drp vel te pe ke mass g posn photoat ball sysball"
try_fstuff "te sysball (endev drp)"
verify
apply (intro exI [where x="0"])
using Os1.lem1 Os2.lem1
diagnose
```

Figure 6.6: Example commands for diagnosing ontologies for a WMS-type of fault, where ‘Os1 and ‘Os2 are two different locales sharing most signature symbols, each with a lemma called ‘lem1’.

6.2.3 Meta-Level Reasoning

The process of diagnosis involves both verifying that the relevant trigger formula is a theorem of the ontologies under consideration and identifying the instantiations of all variables in the trigger formula, e.g., f, *stuff*, and so forth. As already discussed, the generation of the proof goal is done by the system and the user is asked to guide the search of a proof. Once a proof is found, the system then identifies plausible instantiations of the variables in the corresponding trigger formula upon the `diagnose` command. An example execution of WMS up to diagnosis, based on Figure 6.1, is depicted in Figure 6.6.

The WMS, Inconstancy and Reidealisation ontology repair plans detect and repair conflicts among two or more ontologies. They consider terms of the form $f(stuff)$, where both f and *stuff* are polymorphic, higher-order variables. A conflict arises when the value of $f(stuff)$ is unexpectedly different in the ontologies and each of the repair plans proposes a unique repair to resolve the detected conflict. The value of *stuff* is central to the repair process. The WMS repair plan splits *stuff* into visible, invisible and total stuff, so that $f(\textit{stuff})$ refers to the value of f on the visible stuff in one ontology and total stuff in another. The Inconstancy repair plan identifies a new term on which *stuff* depends by observing an unexpected variation in $f(\textit{stuff})$. It adds the identified dependency to *stuff* as a new argument, so that the value of *stuff* varies with the value of it. The Reidealisation repair plan changes the type of *stuff* to one for which there already is an existing law such that the detected contradiction on the value of $f(\textit{stuff})$ can be circumvented.

6.2.3.1 Identifying Variable Instantiations

Unlike Spectrum, the triggering of each WMS, Inconstancy, Reidealisation and Unite is not straightforward: $f(\textit{stuff})$ needs to be instantiated by higher-order matching, which, although it is a decidable problem[5] (Stirling, 2009), f and *stuff* are higher-order polymorphic variables, yielding potentially an unmanageable number of substitutions. We first describe the approach to discovering instantiations of variables in trigger formulae and discuss the mechanism designed to prune the search space to a manageable size in §6.3.

In Isabelle, there are three types of variables: bound, free and schematic variables. Bound variables, e.g., $\forall x.x = x$, and free variables, e.g., $x = x$, are fixed and cannot be instantiated by substitutions. On the contrary, schematic variables, e.g., $?x = ?x$, can be instantiated by the proof process. Since our task is to identify instantiations of f, *stuff* and others, all these variables are expressed as schematic variables. For each ORP, the generated proof goal is matched against a predefined pattern, as outlined in Table 6.1. The schematic variables D_i are not important because we are only interested in the instantiations of variables that contribute to the repair of the ontologies, e.g., f, *stuff*, O_1, O_2, etc. Also, some variables are repeated in the proof goal, e.g., $f(\textit{stuff})$ is mentioned four times and both O_1 and O_2 twice in the proof goal for WMS (6.1).

In order for the approach described so far to behave as intended, all schematic variables in the patterns presented in Table 6.1 are assumed to be polymorphic. However, matching $?f(?\textit{stuff})$ against a sub-expression involving a function hits the most difficult aspect of higher-order unification, because the unification process creates new functions to solve the constraint. If both f and *stuff* are polymorphic, then higher-order

[5] At least up to 4th-order matching, which is more than we need in examples to date.

ORP name	Pattern for higher-order matching
WMS	$\exists v.\ (?O_1 \Longrightarrow ?D_1(?f(?stuff), v) \wedge ?O_2 \Rightarrow ?D_2(?D_3, v)) \vee$ $(?O_1 \Longrightarrow ?D_5(D_6, v))\ \wedge\ (?O_2 \Rightarrow ?D_7(?D_8, v))$
Inconstancy	$\exists c_1, c_2, v_1, v_2.\ (?O_1 \Longrightarrow c_1 \neq c_2 \wedge v_1 \neq v_2) \wedge$ $(?O_2 \Longrightarrow ?f(?stuff) = c_1 \wedge ?v(?b_1) = v_1) \wedge$ $(?O_3 \Longrightarrow ?D_1 = c_2 \wedge ?D_2 = v_2)$
Reidealisation	$(?O_1 \Longrightarrow ?f(?stuff) \neq ?v)\ \wedge\ (?O_2 \Longrightarrow ?D_1)$
Unite	$(?O_1 \Longrightarrow ?D_1(?stuff_1) = ?D_2(?stuff_2)) \wedge (?O_2 \Longrightarrow ?D_3 \neq ?D_4) \wedge$ $(?O_3 \Longrightarrow ?D_5 = ?D_6)$
Spectrum	$?O_1 \Longrightarrow card(?Q) > 1 \wedge card(?P) > 1 \wedge \forall o \in ?D_1.\exists! p \in ?D_2.p(o)$

Table 6.1: Patterns for discovering instantiations of variables in trigger formulae, where $card(S)$ returns the cardinality of the set S; schematic variables $?D_i$ are *don't-cares*, i.e. their values are not important. Each pattern is designed to be matched against one of (6.1), (6.2), (6.3), (6.4) and (6.5).

unification can, potentially, instantiate the type variable to a function type, resulting in a search space that is too large.

In practice, even both $?f$ and $?stuff$ have schematic types, e.g., $?\alpha$ and $?\beta$, respectively, Isabelle does not instantiate schematic type variables to function types during type unification. In other words, Isabelle prevents transforming a schematic type variable into one of some function type, so the arities of schematic variables are fixed for the purpose of higher-order unification. For instance, given the pattern $(?f{:}?\alpha \Rightarrow ?\beta)?stuff$, it will fail to match against the term $(F{:}\mathbb{R} \Rightarrow \mathbb{R} \Rightarrow \mathbb{R})(c)$, because $?\beta$ will not be transformed into a function type. With this restriction in the automation, the search space can be better controlled and managed in most practical uses. However, with our unusual use of Isabelle, we want *stuff* to be both variadic and higher-order. Without modifying Isabelle's unification algorithm, GALILEO dynamically generates a library of type templates for each pattern in Table 6.1, depending on the values of relevant parameters, and executes higher-order matching based on each generated template. If we fix the return type of *stuff* to a constant, then giving *stuff* a maximum arity of 2 and a maximum order of 3 (Definition 2, p.31) yields 183 templates, in which case GALILEO runs matching on the relevant pattern over 183 iterations and stores all substitutions returned. If we allow *stuff* to return first-order unary functions as well, then there are

$((?O_1{:}bool) \Longrightarrow (?f{:}?\alpha \Rightarrow ?\beta)(?stuff{:}?\alpha) \neq (?v{:}?\beta)) \wedge ((?O_2{:}bool) \Longrightarrow (?D_1{:}bool))$
$((?O_1{:}bool) \Longrightarrow (?f{:}(?\alpha_1 \Rightarrow ?\alpha_2) \Rightarrow ?\beta)(?stuff{:}(?\alpha_1 \Rightarrow \alpha_2)) \neq (?v{:}?\beta)) \wedge$ $((?O_2{:}bool) \Longrightarrow (?D_1{:}bool))$
$((?O_1{:}bool) \Longrightarrow (?f{:}(?\alpha_1 \Rightarrow ?\alpha_2) \Rightarrow (?\beta_1 \Rightarrow ?\beta_2))(?stuff{:}(?\alpha_1 \Rightarrow \alpha_2)) \neq$ $(?v{:}(?\beta_1 \Rightarrow ?\beta_2))) \wedge ((?O_2{:}bool) \Longrightarrow (?D_1{:}bool))$

Table 6.2: Three example typed patterns for discovering instantiations of variables for Reidealisation.

366 templates, doubling the previous figure. Three of the 366 typed patterns used for Reidealisation are shown in Table 6.2.

6.3 Search Space Control

As mentioned, because both f and *stuff* are higher-order and polymorphic, there are a vast number of possible instantiations making the unconstrained procedure impractical. Even if the depth of the unification is limited to a small value, e.g., three, a maximum arity of *stuff* of two, and a maximum order of *stuff* of three, the matching procedure still fails to terminate in many test case studies. In order to reduce the number of possible instantiations to a manageable level, we have designed a set of heuristic filters. These filters eliminate physically meaningless instantiations in reference to the Physics context. The size of the filtered set of instantiations is always significantly reduced by several orders of magnitude, while retaining the instantiation that corresponds to the historically accurate and plausible repairs.

6.3.1 Physical Meaningfulness

A physically meaningful instantiation of *stuff* is one that is interpretable within the Physics domain. We argue that not all logically valid or well-formed expressions are interpretable, e.g., $\lambda x.x$, as it is difficult to associate the identity function with a significant and established Physical concept. Perhaps, an abstract or philosophical notion of *self* or *reflection* could be a candidate. Nonetheless, an expression such as $\lambda x.\lambda y.\lambda z.y(z)$ is no doubt even more difficult to translate into the physical world.

Because *stuff* represents the concept that is responsible for producing the detected conflict, limiting the value that *stuff* can take is vital to reducing the huge number of possible instantiations. Much of the focus on analysing meaningfulness is placed on the instantiation of *stuff*, as we need to ensure that it makes sense to repair the instantiation. For instance, if *stuff* was instantiated to the identity function, applying the WMS repair to it, i.e. splitting the identity function into visible and invisible parts, does not allow for a physical interpretation.

6.3.2 Meaningless Instantiation Heuristics

To design appropriate heuristics for filtering out meaningless instantiations, we identify several classes of shapes that undesirable instantiations may exhibit.

Instantiations containing the identity function We argue that repairing an identity function, i.e. $\lambda x.\ x$, is a physically uninterpretable result, so we prohibit *stuff* from being instantiated to it; this is based on the interpretation that an identity function is seen as a form of *reflection*, which is why repairing the notion of reflection would be uninterpretable. However, f can be instantiated to the identity function, as we accept the case in which *stuff* is instantiated to the whole of the target term. We should also consider the case in which the identity function appears as an argument of some function, e.g., $\lambda x.x(\lambda x.x)$. None of the case studies are modelled with the identity function being an argument to some function and we cannot imagine the need for that. So, if f or *stuff* turns out to be instantiated to a term that takes the identity function as an argument, then it must be generated by the unification algorithm. The task of the unification algorithm is, of course, to return logically valid unifiers, without taking physical meaning into consideration. Thus, under the assumption that the original encoding does not apply a function to the identity function, neither f nor *stuff* should be instantiated to a term in which the identity function appears as an argument of some function. Hence, the identity function should, therefore, not appear *anywhere* in the instantiation of *stuff* and only not as a proper sub-expression anywhere in the instantiation of f. We incorporate into GALILEO a heuristic, *H1*, that rejects all instantiations of the form:

$$?stuff \mapsto \ldots(\lambda x.x)\ldots \tag{6.6}$$

and

$$?f \mapsto g(\lambda x.x) \tag{6.7}$$

Instantiations containing schematic variables Neither the instantiations of f nor *stuff* may contain schematic variables in themselves. Schematic variables represent new functions generated by the unification algorithm, which essentially reduces the commitment to the current model of the domain of discourse. Speculating that a new concept exists should not be the effect of the inference, but the result of performing ontology repair by adhering to the appropriate repair rules. So, we design a heuristic, *H2*, to prune away all instantiations of the form:

$$?stuff \mapsto \ldots ?g \ldots \tag{6.8}$$

and

$$?f \mapsto \ldots ?g \ldots \tag{6.9}$$

Instantiations not containing free variables For an instantiation of ?*stuff* to be regarded as being meaningful, it must contain at least one free variable, because a free variable in Isabelle is non-unifiable and represents a concept in the model. It is also important to ensure that the instantiation of ?*stuff* contains at least a term representing a concept, or else repairing an expression without a concept is meaningless. So, we design a heuristic, *H3*, to only accept all instantiations of the form:

$$?stuff \mapsto \ldots k \ldots \tag{6.10}$$

where k is a free variable.

Instantiations with an abstraction as head function symbol If *stuff* is instantiated to a function, we prefer to reject instantiations where the head function symbol is abstracted. Typically, the head function symbol is a constant defined in the ontology, so itself represents a concept in the ontology. Thus, we design a heuristic, *H4*, to filter out all instantiations of the form:

$$?stuff \mapsto \lambda x.\ \ldots x(g) \ldots \tag{6.11}$$

6.3.3 Effectiveness of Heuristics

We briefly discuss the success of these heuristics here. Detailed results are presented in Chapter 8.

The diagnosis process has the ability of producing a collection of plausible diagnoses and higher-order matching $?f(?stuff)$ against an appropriate expression allows creative diagnoses to be produced. In Section §8.3, we explain alternate theories that are physically meaningful and creative. Suppose we match the matter $?f(?stuff)$ against the term

$$TE(Ball, Start(Drop)). \tag{6.12}$$

If the type of $?stuff$ is $?\alpha$, then the unification algorithm does not transform $?stuff$ into a function and effectively applies first-order unification. The resulting unique instantiations are:

$$?f \mapsto \lambda a.a, \qquad ?stuff \mapsto TE(Ball, End(Drop)) \tag{6.13}$$

$$?f \mapsto TE(Ball), \qquad ?stuff \mapsto End(Drop) \tag{6.14}$$

$$?f \mapsto \lambda a.TE(Ball, End(a)), \qquad ?stuff \mapsto Drop \tag{6.15}$$

$$?f \mapsto \lambda a.TE(a, End(Drop)), \qquad ?stuff \mapsto Ball \tag{6.16}$$

The heuristics permit all matches to pass through and all can be given physically meaningful interpretations.

Suppose we have $?stuff : ?\alpha \Rightarrow ?\beta$ and use Isabelle's default parameters for unification; we get over 1,800 matches including:

$$?f \mapsto \lambda a.TE((a(aBall)), End(Drop)), \qquad ?stuff \mapsto \lambda a.a \tag{6.17}$$

$$?f \mapsto \lambda a.TE((aBall), End(Drop)), \qquad ?stuff \mapsto \lambda a.a \tag{6.18}$$

$$?f \mapsto \lambda a.TE((a(?f2(a))), End(Drop)), \qquad ?stuff \mapsto \lambda a.Ball \tag{6.19}$$

$$?f \mapsto \lambda a.a(End(Drop)), \qquad ?stuff \mapsto TE(Ball) \tag{6.20}$$

$$?f \mapsto \lambda a.a(Drop), \qquad ?stuff \mapsto \lambda a.TE(Ball, (End(a)) \tag{6.21}$$

$$?f \mapsto \lambda a.a(Ball), \qquad ?stuff \mapsto \lambda a.TE(a, End(Drop)) \tag{6.22}$$

$$?f \mapsto \lambda a.TE(Ball, a(Drop)), \qquad ?stuff \mapsto \lambda a.End \tag{6.23}$$

H1 rejects both (6.17) and (6.18) because they instantiate $?stuff$ to the identity function; H2 rejects (6.19) for having a schematic-variable, $?f2$, in the instantiation of f. The heuristics filters remove most of the 1,800 matches, leaving only (6.20 - 6.23) and all can be given physically meaningful interpretations.

Suppose we have $?stuff : ?\alpha \Rightarrow ?\beta \Rightarrow ?\gamma$, making $?stuff$ a ternary function. We get over 3,700 matches, but the heuristics H1 - H4 successfully prune away all matches except for (6.24), which emulates the historically accurate solution.

$$?f := \lambda a.a(Ball, End(Drop)), \qquad ?stuff := TE \tag{6.24}$$

The physical meanings of all unfiltered matches are discussed in Chapter 8.

6.4 Post-Identification Diagnosis

As described in §6.2.2.2, §6.2.2.3 and §6.2.2.4, the diagnosis procedures continue after the identification of the instantiation of *stuff* for Inconstancy, Reidealisation and Unite.

For Inconstancy, we need to derive the value of *stuff* from O_1 (4.46, p.60), so it is essential to first obtain the instantiation of *stuff*. The user selects a diagnosis using the command `try_diagnosis ID V`, where `ID` is the diagnosis ID shown in the output and `V` is the value *stuff* is conjectured to take. The subsequent proof obligation is

$$M(p_1, p_2, \ldots, p_m) \Longrightarrow st = V \tag{6.25}$$

where *st* is the instantiation of *stuff* according to the selected diagnosis and V is the value of the `V` argument supplied by the user.

Reidealisation requires to check that $f(stuff\!:\!\tau_2)$ gives consistent values in both O_1 and O_2 (4.33, p.54). This can only be achieved after having identified the instantiation of *stuff* as well. Without knowing the instantiation of *stuff*, we cannot produce quasi-copies of O_1 and O_2 up to *stuff* in O_1 and $stuff'$ in O_2, i.e. ${O_1}'$ and ${O_2}'$ in Reidealisation's trigger formulae. The user selects a diagnosis using the command `try_diagnosis ID`, where `ID` is the diagnosis ID shown in the output. The subsequent proof obligation, in the shape of (4.33), is

$$\exists v.\, M'({p_1}', {p_2}', \ldots, {p_m}') \Longrightarrow st' = v \wedge N'({q_1}', {q_2}', \ldots, {q_m}') \Longrightarrow st' = v \tag{6.26}$$

where st' is the instantiation of $stuff'$ based on the diagnosis selected and M' is the quasi-copy of M up to st in M and st' in M'. Similarly for N'.

Finally, we need the instantiation of *stuff* in order to verify that the defining property of the two stuffs are indeed the same in order to completely trigger Unite (4.65, p.64). Similar to Reidealisation, the user selects a diagnosis using the command `try_diagnosis ID`, where `ID` is the diagnosis ID shown in the output. The subsequent proof obligation, in the shape of (4.65), is

$$\exists v.\, M(p_1, p_2, \ldots, p_m) \Longrightarrow d(st_1) = d(st_2) \Longrightarrow st' = v \tag{6.27}$$

where M is the instantiation of O_1; d is the instantiation of dp; and, st_1 and st_2 are the instantiations of $stuff_1$ and $stuff_2$, respectively.

6.5 Summary

We have realised a powerful form of ontological conflict diagnosis by formulating ontologies as locales. Locales allow for a natural representation of ontologies as contexts, which enables reasoning across multiple instances. Furthermore, conflict diagnosis in GALILEO involves the discovery of the culprit of the detected conflict by utilising higher-order matching, a special case of higher-order unification. Applying a polymorphic variable $?f$ to another, $?stuff$, hits the most difficult aspect of higher-order unification, so we have designed heuristics for preserving only the physically meaningful instantiations. These heuristics are successful in dramatically reducing the options while keeping the desirable ones.

Chapter 7

Mechanising Ontology Repair

7.1 Introduction

Automating the repair of ontologies is a very challenging problem in general, as it involves identifying the axioms and signature elements that require repair and applying appropriate repair operations. The ontology repair technique described in this chapter has been designed to fully automate the repair of axioms and signature that may give rise to the detected ontological fault.

7.2 Ontology Repair in GALILEO

For each ORP, GALILEO performs repair on ontologies by executing the corresponding repair rules. This involves identifying the appropriate elements that require repair and performing the repair on the signature and/or axioms. It is clear from Chapter 4 that the ORPs have vastly different repair behaviours, e.g., WMS introduces a new symbol and a new definition in one ontology whilst renaming occurrences of a certain symbol in another ontology, whereas Spectrum turns unary predicates in an ontology into the return values of a new function. Thus, the main function for performing repair that is implemented in GALILEO is overloaded to behave differently upon the execution of different ORPs. In all cases, a different approach is required for repairing the signature and/or axioms.

7.2.1 Comparison with Axiom-Pinpointing based Repairs

In the belief-revision (Gärdenfors and Rott, 1995) and the DL (Kalyanpur, 2006; Penaloza and Sertkaya, 2009) literatures, simple repair operations are considered sufficient to remove logical contradictions. A common operation is *axiom retraction*, which is to explicitly delete an axiom from an ontology or knowledge base. Typically, the sufficiency of such simple repairs, which is relatively limited, is achieved by the reasonably complex diagnosis machinery used for the discovery of candidate axioms for repair.

The search for axioms that, together, have the given consequence is commonly called *axiom pinpointing* in the belief-revision and DL literatures, where the end result could be Minimal Unsatisfiability Preserving Sub-TBoxes (MUPSs). Axiom pinpointing is typically used for generating justifications that assist to explain the cause of a given fault in ontologies, which helps the user repair the fault later by focusing the manipulation on only those axioms. Typically, axiom pinpointing in DL-related works is implemented by either the *glass-box* or *black-box* approaches, where the former approach involves extending the reasoning algorithm by tracing the axioms used in the derivation of the given consequence whilst the latter queries a reasoner whether a certain subset of the axioms has the given consequence. Axiom pinpointing is inherently difficult, as a given consequence may have exponentially many justifications w.r.t. the size of the TBox (Baader et al., 2007; Penaloza and Sertkaya, 2009). Because axiom pinpointing specifically discovers a subset of candidate axioms, the required repair operation can be relatively simple, e.g., deleting an axiom from a MUPS is sufficient to remove incoherency.

Unlike the typical diagnosis done via axiom-pinpointing as described in the belief revision and DL literatures, GALILEO does not diagnose ontological faults by examining *axioms* but instead by comparing *theorems* that together induce a fault (Chapter 6). We still examine axioms, however, but this is performed during repair and not during diagnosis. We do not find axioms that have the given consequence and then repair them, which is what typical applications of axiom-pinpointing achieve, but instead we transform relevant axioms according to repair operations that are designed to break the *derivation* of the detected fault. Thanks to the way each ORP is designed to limit the search space by outlining the specific characteristics of the proofs and theorems that derive a certain fault in ontologies, our ontology evolution process bypasses the need for tackling axiom pinpointing. Thus, our approach to ontology evolution is completely novel, as the entire process does not involve tracing axioms used in inference and manipulating those axioms, but instead we look at patterns of proof of under- and over-specification and transform axioms that need to be changed based on the suggested problematic concept. Because we work in HOL and the ORPs target various kinds of ontological faults, our repair operations are substantially more sophisticated than those adopted in the related literature, e.g., splitting a concept into parts, adding a new dependency to a concept, reassigning the type of a concept, and so forth.

Each ORP requires axioms containing certain terms to be identified, e.g., parts of the repair rules of WMS (Figure 4.2, p.49) and Inconstancy (Figure 4.5, p.60) require axioms containing occurrences of the instantiation of *stuff* to be identified and then replace those occurrences by a new term. There are a number of techniques for finding such axioms, e.g., syntactic analysis. In GALILEO, we utilise Isabelle's implementation of higher-order unification for this task. As we will illustrate, higher-order unification is particularly natural for identifying axioms that satisfy relatively weak constraints and constructing the repaired formulae. To employ higher-order unification here, we must ensure that the deconstruction of syntax by pattern matching is not disrupted by object-level variables. Suppose we were checking whether a constant `ball` occurs within an axiom containing a variable `X`. Because `X` is a variable, higher-order unification could, theoretically, instantiate it to `ball`, giving a false-positive result. Thus, we must ensure that object-level variables are not instantiated to terms in a pattern. More specifically, we encode patterns using meta-variables and constants, so that the variables used to formulate a pattern are unifiable but those used to formulate object-level sentences are not. That is, we employ higher-order *matching* rather than unification.

Our technique involves formulating the pattern used for matching against the signature and axioms in such a way that a meta-variable is applied to terms that are to be identified from the axioms, if they exist. Suppose we check whether the instantiation of *stuff* occurs within an axiom, which is required by both WMS and Inconstancy, as already mentioned. Suppose *stuff* is instantiated to *Ball* of type Obj. For instance, if the axiom under consideration is

$$Mass(Ball) = 10 \tag{7.1}$$

where $Mass{:}Obj \mapsto \mathbb{R}$, then the pattern for checking whether *Ball* occurs within it is simply

$$?t(Ball) \tag{7.2}$$

where $?t$ is a meta variable of type $Obj \mapsto ?\alpha$ with $?\alpha$ being a schematic type variable. There are two resulting matches

$$?t \quad \mapsto \quad \lambda x{:}\mathbb{R}.Mass(Ball) = 0 \tag{7.3}$$

$$?t \quad \mapsto \quad \lambda x{:}\mathbb{R}.Mass(x) = 0 \tag{7.4}$$

In (7.3), the meta-variable $?t$ is instantiated to a constant function, which would be returned even if *Ball* did not occur within (7.1), so (7.3) is not what we want. The meta-variable $?t$ in (7.4) is instantiated to an unary function with the occurrence of *Ball* λ-abstracted, which is the relevant substitution for our application.

The method described can be naturally generalised to handle more complex scenarios involving more relaxed criteria. For instance, Spectrum requires identifying whether $p(q)$ where $p \in \mathcal{P}$ and $q \in \mathcal{Q}$ occurs within an axiom (4.77, p.68), where $\mathcal{P}$ is a set of all unary predicates and $\mathcal{Q}$ is a set of objects (Figure 4.7, p.68). Note that the axioms that need to be identified here are those containing occurrences of some p applied to some q and not separate occurrences of p and q. Suppose $\mathcal{P}$ contains three unary predicates: *Red*, *Blue* and *Green*, each of type $Obj \mapsto bool$, i.e. $\mathcal{P} := \{Red, Blue, Green\}$, and $\mathcal{Q}$ contains two objects: *Ball* and *Car*, each of type Obj, i.e. $\mathcal{Q} := \{Ball, Car\}$. For instance, suppose the axiom under consideration is

$$Red(Car) \longrightarrow Expensive(Car) \tag{7.5}$$

Because our interest is in checking whether *any* element of $\mathcal{P}$ is applied to *any* element of $\mathcal{Q}$, we match

$$?t(Red, Blue, Green, Ball, Car) \tag{7.6}$$

against the axiom, where $?t$ is of type $(Obj \mapsto bool) \mapsto (Obj \mapsto bool) \mapsto Obj \mapsto Obj \mapsto Obj \mapsto ?\alpha$, taking all elements of $\mathcal{P}$ and $\mathcal{Q}$ as arguments. There are 8 resulting matches

$$?t \quad \mapsto \quad \lambda a,b,c,d,e.\ a(e) \longrightarrow Expensive(e) \tag{7.7}$$

$$?t \quad \mapsto \quad \lambda a,b,c,d,e.\ a(Car) \longrightarrow Expensive(e) \tag{7.8}$$

$$?t \quad \mapsto \quad \lambda a,b,c,d,e.\ Red(e) \longrightarrow Expensive(e) \tag{7.9}$$

$$?t \quad \mapsto \quad \lambda a,b,c,d,e.\ Red(Car) \longrightarrow Expensive(e) \tag{7.10}$$

$$?t \quad \mapsto \quad \lambda a,b,c,d,e.\ a(e) \longrightarrow Expensive(Car) \tag{7.11}$$

$$?t \quad \mapsto \quad \lambda a,b,c,d,e.\ Red(e) \longrightarrow Expensive(Car) \tag{7.12}$$

$$?t \quad \mapsto \quad \lambda a,b,c,d,e.\ Red(Car) \longrightarrow Expensive(Car) \tag{7.13}$$

Although there are numerous matches, only (7.7) and (7.11) indicate that some element of $\mathcal{P}$ is applied to some element of $\mathcal{Q}$ in (7.5). However, (7.11) is the preferred match because (7.7) does not only have the two terms in the antecedent of the axiom abstracted, but also the argument of *Expensive* abstracted as well.

To decide that the set of matches contains a relevant match, i.e. one that is returned only if the axiom contains an occurrence of a certain term, we search through the term instantiated to $?t$ in each match. To do this, we analyse the positions of the λ-abstracted variables in the target term using an implementation of Huet's notion of zippers (Huet, 1997) which represent trees from the perspective of a node. Huet's zippers provide a practical and efficient way to navigate around terms while preserving information about the context. The ORPs require various ways of analysing the target term. For instance, in WMS, we need to ensure that the λ-abstracted variable appears in the body of the expression. To this end, the focus of the zipper is moved to every possible location in search of the corresponding λ-abstracted variable. In comparison, Spectrum requires us to ensure that $p \in \mathcal{P}$ occurs only if it is applied to $q \in \mathcal{Q}$, so the focus of a zipper is moved to each location where an element of $\mathcal{P}$ is abstracted out and check whether the bottom right leaf is an abstraction of an element of $\mathcal{Q}$. In addition, the reverse is applied to ensure that $q \in \mathcal{P}$ only occurs as an argument of $p \in \mathcal{P}$. Both cases are required to be satisfied in order to conclude that an axiom is suitable for a Spectrum-type repair.

7.2.2 Object-Level Repair

Repairing ontologies according to the various ORPs essentially requires locales to be created internally. Even though the result could theoretically be achieved by changing the attributes of an existing locale, it is not robust in practice, because, e.g., tinkering with the current internal state of a theory may cause conflicts within the interactive environment, Isar. So, throughout the implementation of GALILEO, we adhere by the principle of maintaining monotonicity in the working theory, i.e. no existing elements in the working theory are destroyed as a result of executing a GALILEO command, and old, unwanted locales are unlinked from the repaired network.

The effects of ontology repair can be classified into two main classes: signature changes and axiom manipulations.

7.2.2.1 Signature Changes

For the addition of new signature elements, the locale representing the new, repaired ontology receives new locale parameters of the appropriate types; for instance, if a new symbol labelled s of type $\mathbb{R}$ is introduced into the signature, then the set of parameters of the locale representing the new ontology is extended with an element containing the label s and type $\mathbb{R}$. This corresponds to inserting values to the operational part of the specification of a locale in Isar, which is mostly list of the locale parameters. Similarly, if certain signature elements should be retracted from an ontology, then the parameters corresponding to the symbols under consideration are removed from the set of parameters for the construction of the new locale.

7.2.2.2 Axiom Manipulations

The logical part of the specification of a locale define the properties on its parameters. As described in §7.2.1, our approach to axiom pinpointing is capable of selecting a match containing the most relevant instantiation of $?t$, i.e. one that accurately indicates that an axiom indeed contains occurrences of a certain term. Even if the term does not occur in an axiom, $?t$ can be instantiated to a constant function. Thus, the construction of the axioms of a new locale simply involves β-applying the chosen instantiation to $?t$ to appropriate arguments. For instance, in Spectrum, given (7.11), we can repair (7.5) by applying the Spectrum repair by constructing the following term in the new locale

$$(\lambda a,b,c,d,e.\ a(e) \longrightarrow Expensive(Car))$$
$$(\lambda x.Fx = Red, Blue, Green, Ball, Car)$$

Here, we apply $(\lambda a,b,c,d,e.\ a(e) \longrightarrow Expensive(Car))$ to a λ-expression and the last four arguments of $?t$ in the pattern (7.6), which is β-reduced to $(F(Car) = Red) \longrightarrow Expensive(Car)$, as required.

7.2.3 Repair Propagation

All of the case studies covered are modelled using a modular representation, in which ontologies depend on other ontologies forming a network of ontologies. As discussed in §6.2.1, locales may depend on other locales, which allows parameters and axioms to be imported from other locales, so properties on parameters can, consequently, be implicitly stated. For instance, suppose that a locale M has two parameters $p1$ and $p2$ and an axiom $p1 = p2$ and that a locale L extends from M. Locale L may not necessarily introduce new parameters, so it can simply instantiate M using two parameters, e.g., $q1$ and $q2$. Because M is instantiated inside L using $q1$ and $q2$, theorems about $q1$ and $q2$ may be proved using the axiom in M. A ramification of having implicit properties about parameters is that, if the inference for triggering an ORP does not involve axioms that are local in the target locale, repairing that locale does not necessarily make sense, as there is no axiom within that locale available for repair. Thus, allowing such a flexible configuration of ontologies to be given as input introduces the challenge of managing the area of effect of each individual change performed to a locale.

In GALILEO, the effect of a repair is managed by the Repair Propagator (§5.4, p.75), which is responsible for spreading repairs to relevant locales. Every repair operation performed on axioms is propagated to locales from which the current locale extends, so every axiom that is accessible within the current locale is repaired, including those that are implicit w.r.t the target locale. Thus, in order to effect the propagation of repair to axioms, the construction of the repaired network requires an understanding of the dependencies among at least the original locales.

Because the structure of the repaired network of ontologies should imitate that of the original network, the construction of the repaired network also involves an analysis of the dependencies among both the old and new locales. Suppose a locale L depends on a locale M in the original configuration. If M receives a kind of repair that involves inserting a new symbol to the signature of the new M, $\nu(M)$, which is essentially equivalent to adding a new parameter to the specification of $\nu(M)$, then the new dependency between L and $\nu(M)$ must take this new parameter into account, as the new parameter must also occur within the signature of the new L, $\nu(L)$. Without a new parameter declared in $\nu(L)$, the instantiation of $\nu(M)$ within $\nu(L)$ would give rise to a type error, as $\nu(M)$ is essentially a $m+1$ predicate[1], where m is the arity of M; as such, $\nu(L)$ must become a $l+1$ predicate, where l is the arity of L – this is the situation faced by various ORPs, including WMS, Inconstancy and Spectrum.

A locale is created only if the given repair entails changes to the signature and/or the axioms of an existing locale, as GALILEO adopts a minimal approach to the creation of new locales by carrying forward into the repaired network as many of the original ontologies as possible. So, for instance, if only L requires its signature or axioms to be changed, then a new locale, $\nu(L)$, with a dependency on M, is created. This minimal

[1] Recall that locales are implemented as predicates in Isabelle (§6.2.1).

approach is particularly reasonable in the Physics domain, as scientific knowledge is revised only if it is confronted by conflicting evidence.

7.3 Summary

In contrast to belief revision and ontology evolution in DL, GALILEO implements ontology repair without the need for addressing the problem of axiom-pinpointing. The design of each ORP is intended to specify the signature and axioms that should be manipulated in order to resolve a certain kind of fault, whilst still retaining much of the original formalisation. So, the repair operations applied are independent of the inference of the detected fault. The approach adopted involves matching a pattern that is specific to an ORP against each axiom in a target ontology. Such an approach is particularly useful for realising ontology repair, because the instantiation of the meta-variable can be β-applied to some relevant terms, forming the final repaired formula. Because GALILEO accepts dependent locales as input, we also need to manage the dependencies among the old and new locales in order to cope with changes to signatures and implicit axioms.

Chapter 8

Results and Evaluation

8.1 Introduction

The purpose of this chapter is to examine the success of the research and provide evidence to support both the main hypothesis outlined in §1.4, p.5,

> *A few generic, ontology repair plans can account for a large number of historical instances of ontology evolution in the Physics domain*,

and the subordinate hypothesis,

> *A few heuristics enable: (i) substantial control over the size of the search in the space containing solution candidates, which is otherwise unmanageable, and (ii) preservation of only physically relevant solutions.*

We have applied the implementation of our repair plans to the emulation of a small but diverse set of Physics case studies. Each ORP was developed based on some development case studies, from which key design requirements were derived. As already mentioned, the case studies presented in Chapter 4 were used for development and those presented in this chapter were used for evaluation. It is important to base the evaluation of the repair plans on only the case studies and examples that were *not* used to develop the repair plans themselves in order to avoid 'overfitting'. The evaluation of the repair plans is therefore focused on applying them to an independent collection of test case studies[1]. We assess to what extent they (a) create a new ontology that escapes the failures diagnosed in the prior ontology and (b) to what extent this emulates the historical process of ontology evolution or any physically plausible repairs.

[1]Each case study presented in this chapter has been used only for evaluation and not for development

In assessing (b), we take a normative stance, i.e., we are not interested in exactly modelling the historical process, with all its idiosyncracies, false starts, coincidences, etc. Rather, we are content with capturing a 'rational reconstruction' of that history. Since the size of the test set for the repair plans is necessarily measured in tens rather than hundreds or thousands, a quantitative or statistical analysis is inappropriate. Rather, our evaluation methodology is based on discursive analysis of a series of case studies. We look specifically for generality and explanatory power from our repair plans, so we seek diversity in our test set and emergent abstraction from the uniform processing of apparently diverse examples.

In the evaluation, we aim to establish answers to the following questions:

- How broad are the capabilities of each ORP and of the system in its entirety?
- What plausible and implausible repairs does the system hypothesise?
- What are the limitations of the system?

Each of the above is central to determining the success of the project. Firstly, a high generality and potential of scalability of the system means that GALILEO could be extended to solve problems and situations beyond our current scope. An ability to discover new interesting knowledge pushes the system beyond the area of mere resolving ontological conflicts. Finally, limitations of the system should be learnt in order to investigate its applicability and determine its success or to improve it.

8.2 Evaluation Results

In this section, we present the details of only two of the test case studies used to evaluate each repair plan and summaries of others in an attempt to reduce verbosity. For each case study, we describe in depth the modelling approach, the formalisation, and the results from applying the most relevant repair plan.

8.2.1 "Where's My Stuff"

For the WMS repair plan to be triggered (§4.3, p.43), the return value of $f(stuff)$ deduced from one ontology must be different from that deduced from another ontology. Its diagnostic mechanism, therefore, must have assessed a contradiction between the value of $f(stuff)$ and individuated the term *stuff* in the two input conflicting ontologies. The repair performed by WMS amends the signature and axioms of the two conflicting ontologies by redefining the term that instantiates *stuff* in one ontology and renaming occurrences of *stuff* in another.

8.2.1.1 Test Case Study I: The Bouncing-ball Paradox

As described in diSessa (1983), the *bouncing-ball paradox* involves dropping a ball above ground and predicting the amount of total energy when it impacts with the ground. Some Physics students took a (wrong) definition of total energy, which is defined as the sum of kinetic energy, which is initially zero as it is held stationary, and potential energy, which is initially greater than zero as the ball is held above ground. By the law of conservation of energy, the final amount of the total energy can be computed. The observed final amount of total energy is zero, because there is no kinetic energy and no potential energy due to gravity. The paradox is exactly the contradiction between the initial and final amounts of total energy of the ball. It is elastic but the definition of total energy applied is the one for particles *without* extent.

When a ball is lifted off the ground, work is being done to the ball, increasing its *potential* energy. There are various specific types of potential energies, each dependent on the kind of force that performs the work. In the case of lifting a ball, the external force works against the force of the earth's gravity, with which the associated potential energy is defined as:

$$E_p = m \times g \times h \tag{8.1}$$

where

- E_p is the amount of potential energy of the object relative to its being on the earth's surface
- m is the mass of the object
- g is the acceleration due to gravity
- h is the altitude of the object.

To produce a logical formulation, we do replace variables in an equation by function applications:

$$\forall o,t.\ PE(o,t) = Mass(o,t) \times G \times Height(o,t) \tag{8.2}$$

where

- $(PE{::}Obj \Rightarrow Mom \Rightarrow \mathbb{R})(o,t)$ returns the amount of potential energy of the object o at a time t relative to its being on the earth's surface

- $(Mass{::}Obj \Rightarrow Mom \Rightarrow \mathbb{R})(o,t)$ returns the mass of the object o at a time t[2]

- $(G{:}\mathbb{R})$ is the acceleration due to gravity

- $(Height{::}Obj \Rightarrow Mom \Rightarrow \mathbb{R})(o,t)$ returns the altitude of the object o at a time t.

When the ball is dropped, the ball begins to move toward the ground due to Earth's gravitational pull. Assuming the ball is dropped vertically without rotation, its stored potential energy begins to be converted to *kinetic* energy due to its motion, which is given by the equation:

$$E_k = \frac{1}{2} m \times v^2 \tag{8.3}$$

where

- E_k is the kinetic energy of the object in consideration due to the motion

- m is the mass of the object

- v is the velocity of the object.

Similar to earlier, we can replace all variables by functions to produce the following logical formulation:

$$\forall o,t.\ KE(o,t) = \frac{1}{2} Mass(o,t) \times Velocity(o,t)^2 \tag{8.4}$$

where

- $(KE{::}Obj \Rightarrow Mom \Rightarrow \mathbb{R})(o,t)$ returns the amount of kinetic energy of object o at a time t due to motion

- $(Mass{::}Obj \Rightarrow Mom \Rightarrow \mathbb{R})(o,t)$ returns the mass of object o at a time t

- $(Velocity{::}Obj \Rightarrow Mom \Rightarrow \mathbb{R})(o,t)$ returns the velocity of object o at a time t.

[2]We try to reuse as many of the symbols as possible in our representation. A generic representation of mass opens up the possibility of faulty or misleading physics, but itself is a strength rather than a weakness given our application since we do want to experiment with various ontological faults and repairs.

Throughout the fall of the ball, the total amount of energy in it is constant; this is known as the *law of conservation of energy*. Energy is always conserved over time. It may change forms, but the total amount of energy in a closed system is conserved. Here, we consider the ball alone is a closed system. The law of conservation of energy can be defined as:

$$\sum_{i}^{n} E_{0_i} = \sum_{i}^{m} E_{f_i} \tag{8.5}$$

where E_{0_i} is the amount of a particular type of energy out of n many types available at the beginning of some interval and E_{f_i} is the amount of a particular type of energy out of m many types available at the end of the interval. Since energy may change forms, n and m are not necessarily the same.

Suppose the experiment was conducted by taking a series of photos of the ball separated by a positive constant interval $\delta{:}\mathbb{R}$ while it is dropped. The ontologies containing the theory and instance data are shown in Figure 8.1. Note that *Obs*, the observation ontology, uses a different language, as *Pos* and *Photo* are not part of the language of *Pred*, the predictive theory – they share only *Ball*, *Height* and *Drop*. The function $(Pos{:}Obj \Rightarrow Photo \Rightarrow \langle\mathbb{R},\mathbb{R}\rangle)(o,p)$ returns the position of object o according to the photo object p. The position is represented as a pair of real numbers, where the first corresponds to the number of units in the x-axis and the second corresponds to the number of units in the y-axis.

Because we work with factorised representations when reasoning with heterogeneous ontologies (§5.6, p.85), there must exist an ontology within the factorised representation of *Pred*, $\mathcal{F}(Pred)$, such that the merge between *Obs*, the bridging ontology, and this ontology does not induce inconsistency. Suppose this ontology is $Pred_f$:

$$
\begin{aligned}
Ax(Pred_f) \quad ::= \quad & \{ \\
& \forall o{:}Obj, t{:}Mom. KE(o,t) = \frac{1}{2} Mass(o,t) \times \\
& Velocity(o,t)^2, && (8.16) \\
& \forall o{:}Obj, t{:}Mom. PE(o,t) = Mass(o,t) \times G \times \\
& Height(o,t), && (8.17) \\
& \forall t{:}Mom.\ TE(Ball,t) = KE(Ball,t) + PE(Ball,t), && (8.18) \\
& Mass(Ball, Start(Drop)) > 0, && (8.19) \\
& G > 0 && (8.20) \\
& \}
\end{aligned}
$$

$$
\begin{aligned}
Ax(Pred) \supseteq \{ & \\
& \forall o{:}Obj, t{:}Mom. KE(o,t) = \frac{1}{2} Mass(o,t) \times & (8.6)\\
& Velocity(o,t)^2, \\
& \forall o{:}Obj, t{:}Mom. PE(o,t) = Mass(o,t) \times G \times & (8.7)\\
& Height(o,t), \\
& \forall t{:}Mom.\ TE(Ball,t) = KE(Ball,t) + PE(Ball,t), & (8.8)\\
& \forall o{:}Obj,\ t_1,t_2{:}Mom.\ TE(o,t_1) = TE(o,t_2), & (8.9)\\
& Mass(Ball, Start(Drop)) > 0, & (8.10)\\
& G > 0, & (8.11)\\
& Height(Ball, Start(Drop)) > 0, & (8.12)\\
& Velocity(Ball, Start(Drop)) = 0 & (8.13)\\
& \} \\
Ax(Obs) \supseteq \{ & \\
& Pos(Ball, Photo(End(Drop) - 1)) = (0,0) & (8.14)\\
& Pos(Ball, Photo(End(Drop))) = (0,0) & (8.15)\\
& \}
\end{aligned}
$$

where *Pred* is the predictive ontology containing the theory and initial observations and *Obs* contains measurements collected from a series of photographs.

Figure 8.1: Axiomatisation of a heterogeneous representation of the bouncing-ball paradox.

As the axioms (8.16 - 8.20) make a subset of those of *Pred*, $Pred_f$ is guaranteed to be a node in $\mathcal{F}(Pred)$.

To link the seemingly disparate terms together, the bridging axiom that needs to be encoded in the bridging ontology, O_b, is:

$$\begin{aligned}
&\forall p : Particle,\ t : Mom, v : \mathbb{R}. \\
&\quad Obs \vdash \frac{snd(Pos(p, Photo(t-\delta))) - snd(Pos(p, Photo(t)))}{\delta} = v \longleftrightarrow \\
&\quad Pred_f \vdash Velocity(p,t) = v
\end{aligned} \tag{8.21}$$

$$\begin{aligned}
&\forall p : Particle,\ t : Mom, v : \mathbb{R}. \\
&\quad Obs \vdash snd(Pos(p, Photo(t))) = v \longleftrightarrow Pred_f \vdash Height(p,t) = v
\end{aligned} \tag{8.22}$$

where (8.21) relates the notion of velocity in *Pred* to the symbols in *Obs* by stating that the velocity of an object at some moment t is equal to the difference between the vertical positions of the object captured in the photographs taken at t and $t-1$ many δ[3]; (8.22) relates the notion of height in *Pred* to the symbols in *Obs* by stating that the height of an object at some moment t is equal to the vertical positions of the object captured in the photographs taken at t; and, *snd* returns the second element in a tuple. The ontology $Pred_f$ refers to an ontology in $\mathcal{F}(Pred)$ containing certain axioms (8.16 - 8.20). This assertion enables the inference of properties about concepts that are not expressible in *Obs*, e.g., velocity, by using facts expressed in a different language, e.g., positions of an object in two photos, and vice versa.

Thus, from these ontologies, the paradox described by diSessa (1983) can be formulated as follows:

$$Pred \vdash TE(Ball, End(Drop)) > 0 \tag{8.23}$$

$$Obs \oplus O_b \oplus Pred_f \vdash TE(Ball, End(Drop)) = 0. \tag{8.24}$$

Equation 8.23 is derivable because we can first apply transitivity to obtain two subgoals; one can be resolved with (8.9) in Figure 8.1, while the other,

$$TE(Ball, Start(Drop)) > 0, \tag{8.25}$$

remains open. We can rewrite the LHS of the equation first with (8.8), and then with

[3] If the full ontology contains an assertion that velocity is the derivative of height, we could work with approximations instead rather than equality and have $\ldots \frac{snd(Pos(p,Photo(t-\delta)))-snd(Pos(p,Photo(t)))}{\delta} \approx v \ldots$ in the bridging axiom.

(8.6) and (8.7) to give:

$$\frac{1}{2}Mass(Ball,Start(Drop)) \times Velocity(Ball,Start(Drop))^2 + \\ Mass(Ball,Start(Drop)) \times G \times Height(Ball,Start(Drop)) > 0. \quad (8.26)$$

We can resolve (8.26) by rewriting with (8.10), (8.11), (8.12), and (8.13). The derivation of (8.24) is similar, as it can be rewritten first with (8.18), and then with (8.16) and (8.17). We can then rewrite using the encoded bridging axioms (8.21, 8.22) and then with (8.14, 8.15).

The paradox can trigger the WMS repair plan with the following substitution:

$$\{Pred/O_1, Obs \oplus O_b \oplus Pred_f/O_2, \\ TE/stuff,\ \lambda x.\ x(Ball,End(Drop))/f, 0/v\} \quad (8.27)$$

with the type variables instantiated by the following substitution:

$$\{Obj \Rightarrow Mom \Rightarrow \mathbb{R}/\tau,\ \mathbb{R}/\tau'\} \quad (8.28)$$

where the function *TE* is regarded as *stuff*, which requires a new definition to be resulted from the repair.

Similar to the previous case study, the substitution (8.27) and (8.28) is only one of several that can trigger WMS, but it is the most interesting one as it allows the historical solution to be produced. We will explore alternative repairs that are produced by the remaining substitutions in §8.3.

To effect the repair we will write the visible stuff as TE_{part} and invisible stuff as EE, in anticipation of their intended meanings, where TE_{part} and EE can be read as the total energy for particles and the elastic energy for particles with extent, respectively. These choices instantiate (4.15) to:

$$\forall o{:}Obj, t{:}Mom.\ EE(o,t) \quad = \quad TE(o,t) - TE_{part}(o,t) \quad (8.29)$$

which means that the amount of elastic energy is defined as the difference between the amount of total energy in the object and the amount total energy for particles, i.e. kinetic and potential energies. The result is to modify the set of axioms in *Pred* and augment it with (8.29). All occurrences of *TE* in the axioms of the combined ontology to which O_2 is instantiated are renamed to TE_{part}. Because $Pred_f$ is reachable from other ontologies in $\mathcal{F}(Pred)$, the renaming is propagated to every ontology in

$\mathcal{F}(Pred)$, as expected. The set of axioms resulted from the repair is shown in Figure 8.2. In the repaired ontologies, the detected contradiction is now resolved:

$$\begin{aligned} \nu(Pred) &\vdash TE(Ball, End(Drop)) > 0 \\ &\quad \forall o{:}Obj, t{:}Mom.\ EE(o,t) = TE(o,t) - TE_{part}(o,t) \\ \nu(Obs) \oplus \nu(O_b) \oplus \nu(Pred_f) &\vdash TE_{part}(Ball, End(Drop)) = 0. \end{aligned}$$

We could interpret the repaired axioms to mean that the observation values were based only on the consideration that the ball is a particle without extent, which resulted in lower than expected values.

$$\begin{aligned} Ax(\nu(Pred)) \supseteq \{ & \\ & \forall t{:}Mom.\ EE(Ball,t) = TE(Ball,t) - TE_{part}(Ball,t), \\ & \forall o{:}Obj, t{:}Mom. KE(o,t) = \frac{1}{2}Mass(o,t) \times \\ & Velocity(o,t)^2, \\ & \forall o{:}Obj, t{:}Mom. PE(o,t) = Mass(o,t) \times G \times \\ & Height(o,t), \\ & \forall t{:}Mom.\ TE(Ball,t) = KE(Ball,t) + PE(Ball,t), \\ & \vdots \\ \} & \\ Ax(\nu(Pred_f)) ::= \{ & \\ & \forall o{:}Obj, t{:}Mom. KE(o,t) = \frac{1}{2}Mass(o,t) \times \\ & Velocity(o,t)^2, \\ & \forall o{:}Obj, t{:}Mom. PE(o,t) = Mass(o,t) \times G \times \\ & Height(o,t), \\ & \forall t{:}Mom.\ TE_{part}(Ball,t) = KE(Ball,t) + PE(Ball,t), \\ & \vdots \\ \} & \end{aligned}$$

Figure 8.2: Summary of the repaired axiomatisation of the bouncing-ball paradox.

Note that the new definition (8.29) quantifies over all entities in the domain, i.e. all possible objects, but elasticity may not be important to the calculation of energy of certain kinds of objects. For instance, there are objects that exhibit other kinds of potential energies, e.g., electric potential energy and chemical potential energy. That said, we can imagine WMS to perform repair again when a conflict arises from experimenting with these kinds of objects in a similar fashion, i.e. creates a new definition representing the corresponding new kind of energy. Nonetheless, WMS has successfully introduced a new kind of energy, which is what is required in this case study.

The complete details of the Isabelle theory file for the bouncing-ball case study is shown in B.1.

8.2.1.2 Test Case Study II: Dark Matter

The evidence for dark matter comes from various sources, for instance, from an anomaly in the orbital velocity of stars in spiral galaxies identified by Rubin et al. (1980). Given the observed distribution of mass in these galaxies, we can use Newtonian Mechanics to predict that the orbital velocity of each star should be inversely proportional to the square root of its distance from the galactic centre (called its *radius*), i.e.

$$v_o = \sqrt{\frac{GM}{r}} \tag{8.30}$$

where

- v_o is the orbital velocity of the object
- G is the gravitational constant
- M is the mass around which the object is orbiting
- r is the distance between the object and the centre of the mass around which the object is orbiting.

Note that (8.30) is an approximation of the true orbital velocity where the mass of the object under consideration is negligible when compared to M. To produce a logical formulation of (8.30), we replace variables by function applications:

$$\forall o_1, o_2, t.\ OrbVel(o_1, o_2, t) = \sqrt{\frac{G \times Mass(o_2, t)}{Radius(o_1, o_2, t)}} \tag{8.31}$$

where

- $(OrbVel :: Obj \Rightarrow Obj \Rightarrow Mom)(o_1, o_2, t)$ returns the velocity of an object, o_1, orbiting around another object, o_2, at a time t

- $(G{::}\mathbb{R})$ is the gravitational constant
- $(Mass{::}Obj \Rightarrow Mom)(o,t)$ returns the mass of the object o at a time t[4]
- $(Radius{::}Obj \Rightarrow Obj \Rightarrow Mom)(o_1,o_2,t)$ returns the distance (radius) between the centre of object o_1 and the centre of another object o_2 around which o_1 is orbiting at a time t.

As most of the observed stars orbit around the centre of a galaxy, we need to compute the mass of the galaxy as required by the RHS of (8.31). We assume that the mass of a galaxy can be estimated by summing the mass of all matter that substantially contribute to the mass of the galaxy, i.e.

$$Mass(g,t) ::= \sum_{s \in g} Mass(s,t) \tag{8.32}$$

where *Glxy* is the type of galaxies[5]

Note that the function *Mass* is overloaded and (8.32) is defined for computing the mass of galaxies.

However, observations on the rotational velocity of several galaxies show the orbital velocity of constituent stars to be roughly constant and independent of their radius[6]. Figure 8.3 illustrates the predicted and actual rotation curves. In order to account for this discrepancy, it is hypothesised that galaxies also contain a halo of, so called, *dark matter*, which is invisible to our radiation detectors, such as telescopes, because it does not radiate, so can only be measured indirectly. Since the predicted curve is a plot of the orbital velocity of, e.g., stars and gas clouds, that constitute the galaxy, we can define the graph object as:

$$GraphA(g,t) ::= \lambda s \in g.\ \langle Radius(s,g,t), OrbVel(s,g,t) \rangle \tag{8.33}$$

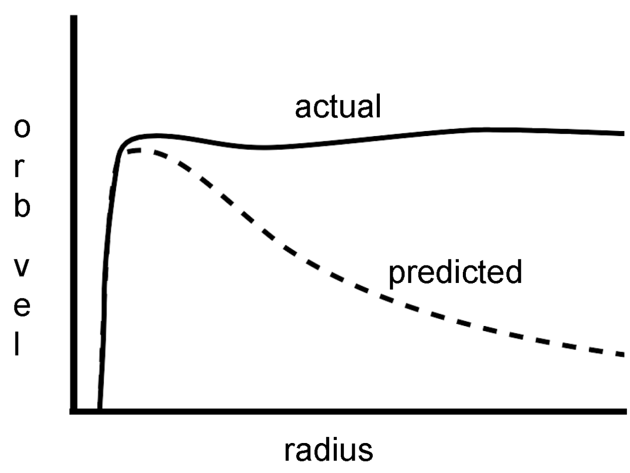
where the graph is expressed as a function over all constituent objects. The graph itself depends on the galaxy and the time moment considered, assuming its rotational velocity could be time dependent.

Suppose the predicted rotation curve is based on Newtonian Mechanics and the observed rotation curve was made by plotting the orbital velocity of various objects along

[4] The function is assumed to be compounded with the mathematics required to compute the mass of all stars at the radius of o at t.

[5] The *Glxy* type is synonymous with a set of matter. The elements of the set are those masses that substantially contribute to the mass of the underlying galaxy.

[6] The rotational velocity of a galaxy can be derived from the orbital velocity of its constituent stars, and vice versa.



This diagram is taken from `http://en.wikipedia.org/wiki/Galaxy_rotation_problem`. *The x-axis is the radii of the stars and the y-axis is their orbital velocities. The dotted line represents the predicted graph and the solid line is the actual graph that is observed.*

Figure 8.3: Predicted *vs* Observed Stellar Orbital Velocities

the galactic plane against the distance from the galactic centre. Typically, the rotational velocity of galaxies at different radii is derived by examining the resulting Doppler effect, which is the shift in frequency of a wave for an observer due to the relative motion of the source of the wave. For instance, if a light source moves away from an observer, the observed colour shifts to longer wavelengths, e.g., red, and if it moves closer to an observer, the colour shifts to shorter wavelengths, e.g., blue.

In Figure 8.4

- $glxy101{:}Glxy$ is the galaxy considered, which is made up of *star001*, *star002*, etc.;
- $(GraphA{:}Glxy \Rightarrow Mom \Rightarrow Matter \Rightarrow (\mathbb{R},\mathbb{R}))(g,t)$ is the rotation curve of galaxy g at moment t plotted based on the predictive theory;
- $(GraphB{:}Glxy \Rightarrow Mom \Rightarrow Matter \Rightarrow (\mathbb{R},\mathbb{R}))(g,t)$ is the rotation curve of galaxy g at moment t plotted based on rotational velocities, calculated using empirical measurements;
- $(RotVel{:}Obj \Rightarrow Glxy \Rightarrow Mom \Rightarrow \mathbb{R})(o,g,t)$ returns the rotational velocity of object o in galaxy g at a time t;

$$Ax(Pred) \supseteq \{$$

$$OrbVel(o_1, o_2, t) = \sqrt{\frac{G \times Mass(o_2, t)}{Radius(o_1, o_2, t)}}, \tag{8.34}$$

$$Mass(g{:}Glxy, t) = \sum_{s \in g} Mass(s, t) \tag{8.35}$$

$$GraphA(g, t) = \lambda s \in g.\ \langle Radius(s, g, t), OrbVel(s, g, t) \rangle \tag{8.36}$$

$$G = 6.673 \times 10^{-11} \tag{8.37}$$

$$glxy101 = \{star001, star002, \ldots\} \tag{8.38}$$

$$Mass(star001, t) = 1 \times 10^{8},\ \ldots \tag{8.39}$$

$$Radius(star001, glxy101, t) = 1 \times 10^{10},\ \ldots \tag{8.40}$$

$$\}$$

$$Ax(Obs) \supseteq \{$$

$$RotVel(o, g, t) = \frac{RadVel(o, g, t) - ObsVel(o, g, t)}{sin(inclination(g, t))}, \tag{8.41}$$

$$RadVel(o, g, t) = c \times Redshift(o, g, t) \tag{8.42}$$

$$Redshift(o, g, t) = \frac{WavelengthObs(o, g, t)}{WavelengthEmit(o, g, t)} - 1 \tag{8.43}$$

$$GraphB(g, t) = \lambda s \in g.\ \langle Radius(s, g, t), RotVel(s, g, t) \rangle \tag{8.44}$$

$$inclination(glxy101, t) = 1, \tag{8.45}$$

$$WavelengthObs(star001, glxy101, t) = 663,\ \ldots \tag{8.46}$$

$$WavelengthEmit(star001, glxy101, t) = 656,\ \ldots \tag{8.47}$$

$$\ldots$$

$$\}$$

where the ontology *Pred* formalises the predictive theory, whereas *Obs* contains the equations needed for converting from measurements performed using a spectroscopic method to rotational velocity.

Figure 8.4: Axiomatisation of a heterogeneous representation of the discovery of dark matter.

- $(RadVel{:}Obj \Rightarrow Glxy \Rightarrow Mom \Rightarrow \mathbb{R})(o,g,t)$ returns the radial velocity of object o in galaxy g at a time t, which is the velocity of the object in the direction of the line-of-sight;

- $(ObsVel{:}Obj \Rightarrow Glxy \Rightarrow Mom \Rightarrow \mathbb{R})(o,g,t)$ returns the observed velocity of o in galaxy g at a time t as observed by the observer

- $(inclination{:}Glxy \Rightarrow Mom \Rightarrow \mathbb{R})(g,t)$ returns the angular distance of the orbital plane of g from the equator;

- $c{:}\mathbb{R}$ is the speed of light;

- $(Redshift{:}Obj \Rightarrow Glxy \Rightarrow Mom \Rightarrow \mathbb{R})(o,g,t)$ is for calculating the amount of wavelength shifted, depending on the values of the observed wave length;

- $(WavelengthObs{:}Obj \Rightarrow Glxy \Rightarrow Mom \Rightarrow \mathbb{R})(o,g,t)$ returns the amount of wavelength shifted; and,

- $(WavelengthEmit{:}Obj \Rightarrow Glxy \Rightarrow Mom \Rightarrow \mathbb{R})(o,g,t)$ returns the wavelength emitted.

RotVel is not part of the language of the predictive theory, but we know that rotational velocity of a galaxy at a given distance can be approximated to be the orbital velocity of the body at that distance. Similar to the previous case study, because we use factorised representations when reasoning with heterogeneous ontologies, there must exist an ontology within $\mathcal{F}(Pred)$ such that the merge between *Obs*, the bridging ontology and this ontology does not induce inconsistency. Suppose this ontology is $Pred_f$:

$$
\begin{aligned}
& Ax(Pred_f) \supseteq \{ \\
& \quad GraphA(g,t) = && (8.48) \\
& \quad \lambda s \in g.\ \langle Radius(s,g,t), OrbVel(s,g,t) \rangle \\
& \quad glxy101 = \{star001, star002, \ldots\} && (8.49) \\
& \quad \}
\end{aligned}
$$

The axioms (8.48 - 8.49) make a subset of those of *Pred*, so $Pred_f$ is guaranteed to be a node in $\mathcal{F}(Pred)$.

To formalise this association between unfamiliar symbols in *Pred* and *Obs*, the bridging axiom required to be encoded in the bridging ontology, O_b, is:

$$\begin{aligned} &\forall r, v{:}\mathbb{R}, g{:}Glxy, t{:}Mom. \\ &\quad Pred \vdash OrbVel(o,g,t) = v \longleftrightarrow \\ &\quad Obs \vdash RotVel(o,g,t) = v. \end{aligned} \tag{8.50}$$

These ontologies alone are in fact insufficient to reason about the shapes of two rotation curves. Even if the ontologies are large enough to contain the data about each and every object in a galaxy, the resulting plots would not be continuous curves. We, therefore, produced "helper" ontologies for reasoning about shapes of graphs. In particular, we provide a way to compare two curves by examining their gradients, cutoff points, intersections, etc. For instance, the predicted and observed curves in Figure 8.3 overlap until a cutoff point. Up to this point, the two curves have the same gradient. From here onward, the predicted curve monotonically declines whereas the observed curve becomes almost flat. Thus, the gradient of the predicted curve monotonically decreases, whereas that of the observed curve becomes roughly constant. We can further draw inference about other properties of a curve, given descriptions about the gradient of various segments of it. For example, if two curves start at the same point but one is parallel to the x-axis whereas the other one declines, then the former must return a larger y-value than the latter for every x-value; one could, therefore, conclude the former to be "larger" than the latter, i.e. for two graphs $g_1{:}\mathbb{R} \Rightarrow \mathbb{R}$ and $g_2{:}\mathbb{R} \Rightarrow \mathbb{R}$ and a starting point p

$$\begin{aligned} &\forall x_1, x_2{:}\mathbb{R}.\ x_1 > p \wedge x_2 > p \wedge x_1 \neq x_2 \wedge g_1(p) = g_2(p) \wedge \\ &\quad Gradient(x_1,x_2,g_1) < Gradient(x_1,x_2,g_2) \longrightarrow g_1 < g_2 \end{aligned}$$

where $(Gradient{:}\mathbb{R} \Rightarrow \mathbb{R} \Rightarrow (\mathbb{R} \Rightarrow \mathbb{R}) \Rightarrow \mathbb{R})(x_1,x_2,g)$[7] returns the rate of increase/decrease between two points on g with values x_1 and x_2 on the x-axis. Note that the $<_\tau$ operator is polymorphic, which enables us to compare both numbers and graphs.

From the ontologies in Figure 8.4 along with knowledge for making graph comparisons, bridging axiom and $\mathcal{F}(Pred)$, the discrepancy between the curves in Figure 8.3 can be formulated as follows:

$$Pred \vdash GraphA(glxy101,t_1) < UpFlat \tag{8.51}$$

$$Pred_f \oplus Obs \oplus O_b \vdash GraphA(glxy101,t_1) = UpFlat \tag{8.52}$$

[7]Without being limited to handling only graphs that take reals, the gradient function implemented in GALILEO is polymorphic. For the convenience of explanation, we assume it only handles reals here.

where t_1 is a time moment during the relevant observation and *UpFlat* is a graph that is defined to have the same shape as the observed curve in Figure 8.3, i.e. positive gradient between the origin and a cutoff point (up) and a constant gradient thereafter (flat). We can prove (8.51) by rewriting the LHS first with (8.36) and then with (8.34), (8.35), (8.37), (8.38), (8.39) and (8.40). Since *Pred* contains the value of G, the values of the mass of stars and the value of the radius between each star and its galactic centre, we can plot the predicted rotation curve, $GraphA(glxy101, t_1)$. With the knowledge for making graph comparisons, we infer that none of the points on $GraphA(glxy101, t_1)$ is higher than those on *UpFlat* but some of the points are lower than those on *UpFlat*. For the rotation curve based on the empirical observations, we can prove (8.52) by rewriting the LHS first with (8.48), then with the encoded bridging axiom (8.50), and then with (8.41 - 8.43), (8.46) and (8.47). The ontology *Obs* contains instance data, so we can plot the value of each point on $GraphA(glxy101, t_1)$ by deriving values of rotational velocity from observations. We can infer that *GraphA* and *GraphB* are the same by (8.50), (8.36) and (8.44), the predicted and observed curves are expected to have also the same shape, i.e. the shape of *UpFlat*.

We can instantiate the trigger formulae of WMS (4.13 - 4.14, p.49) with the following:

$$\begin{aligned}&\{Pred \oplus Obs \oplus O_b/O_1,\\ &Pred/O_2, glxy101/stuff,\ \lambda x.\ GraphA(x,t_1)/f, UpFlat/v\}\end{aligned} \tag{8.53}$$

with the type variables instantiated by the following:

$$\{Glxy/\tau,\ Matter \Rightarrow (\mathbb{R},\mathbb{R})/\tau'\}$$

where the galaxy $glxy101$ is regarded as *stuff*, which is to be redefined by the repair. Unlike the previous case study, *stuff* is instantiated to an argument of the dominant function rather than the dominant function itself. The substitution (8.53) and (8.54) is only one of several that can trigger WMS, but it is the most interesting one as it allows the historical solution to be produced. We will investigate other plausible repairs in §8.3.

To effect the repair we will write the visible stuff as $glxy101_{vis}$ and the invisible stuff as $glxy101_{invis}$, in anticipation of their intended meanings, where $glxy101_{vis}$ and $glxy101_{invis}$ can be read as the part of the galaxy that emits light or other radiation and the part of galaxy that does not, respectively. These choices instantiate (4.15) to:

$$glxy101_{invis} = glxy101 - glxy101_{vis} \tag{8.54}$$

which means that the entirety of the galaxy under consideration is defined to be composed of both *visible* and *invisible* parts. The new concept created by WMS, $glxy101_{invis}$, corresponds to the hypothetical matter, dark matter. Old axioms need to be adjusted in order to be expanded with the new definition (8.54) and GALILEO selects the destination ontology for the new definition by executing the ontology selection algorithm on the ontologies involved according to the instantiation of O_1. The result is to modify the set of axioms in *Obs*, of which the signature contains the declaration of $glxy101$, and augment it with (8.54). All occurrences of $glxy101$ in the axioms of the combined ontology to which O_2 is instantiated are renamed to $glxy101_{vis}$ – unlike the previous example, this includes only those within *Pred*. Because O_1 receives only a new definition as repair, even though $Pred_f$ is reachable from other ontologies in $\mathcal{F}(Pred)$, only *Obs* receives the change. Figure 8.5 shows the resulting set of axioms. In the repaired ontologies, the detected contradiction is now resolved:

$$\begin{aligned} Pred &\vdash GraphA(glxy101_{vis}, t_1) < UpFlat \\ Pred \oplus Obs \oplus O_b &\vdash GraphA(glxy101, t_1) = UpFlat \\ &\vdash glxy101_{invis} = glxy101 - glxy101_{vis}. \end{aligned}$$

In essence, the repaired axioms could be interpreted to mean that the predicted values were in fact made only on the emitting regions of the galaxy and not on the galaxy as a whole, as there were hidden regions that were not considered by the predictive theory. This solution precisely emulates the discovery of dark matter.

8.2.1.3 Other Test Case Studies

In addition to the case studies presented in (§8.2.1.1) and (§8.2.1.2), we have also evaluated WMS over a diverse range of other case studies. These include:

- *Friction*: Newton's first law of motion works well for astronomical objects, but in a terrestrial environment, objects travelling in constant velocity with no apparent applied force eventually come to rest. This could be interpreted as a discrepancy between the predicted and observed graphs of velocity plotted against time. The WMS patch is to invent an invisible force: friction.

- *Discovery of Neutrons*: Rutherford had theorised that nuclei were composed of protons and that the number of protons determined the atomic number. However, he later discovered a disparity between an element's atomic number and

$$
\begin{aligned}
Ax(\nu(Pred)) \supseteq \{ & \\
& G = 6.673 \times 10^{-11} \\
& glxy101_{vis} = \{star001, star002, \ldots\} \\
& Mass(star001, t) = 1 \times 10^{8}, \ \ldots \\
& \} \\
Ax(\nu(Obs)) \supseteq \{ & \\
& glxy101_{invis} = glxy101 - glxy101_{vis} \\
& inclination(glxy101, t) = 1, \\
& WavelengthObs(star001, glxy101, t) = 663, \ \ldots \\
& WavelengthEmit(star001, glxy101, t) = 656, \ \ldots \\
& \}
\end{aligned}
$$

Figure 8.5: Summary of the repaired axiomatisation of the discovery of dark matter.

its atomic mass. The repair devised was to introduce some "invisible stuff" and postulate the existence of neutrons – a new kind of massive sub-atomic particle – to account for the missing weight, which Chadwick later experimentally discovered.

- *Discovery of Neutrinos*: In 1930, Pauli addressed the failure of the law conservation of energy on the subatomic level by introducing a new sub-atomic particle. Neutrons, which were discovered two years later, were too massive for balancing the equations. The WMS type of repair undertook was to postulate "little neutral ones", or neutrinos.

8.2.1.3.1 Friction Suppose the experiment involves measuring the velocity of an object over several episodes. In each episode, the same amount of force is applied to it, but over different durations, e.g., 10N for one second, 10N for two seconds, and so forth. Because of friction, the observed velocity is always lower than that predicted. We can instantiate the WMS repair plan with the following substitution:

$$
\{Pred/O_1, Obs/O_2, Force/stuff, \lambda x, y.ForceToVel(y, x(y))/f, \lambda x, y.\langle y, Vel(x)\rangle/v\} \tag{8.55}
$$

where *Pred* and *Obs* are the predictive and observation ontologies, respectively; $Force(o)$ returns the amount of force being applied to an object o; $ForceToVel(o,a,d)$ returns a graph plotting the velocity some object o travels if a force of the amount a is applied for d seconds; and, $Vel(o)$ is measure the velocity of an object o. *ForceToVel* essentially integrates the acceleration once to find the velocity. The repair is to split *Force* into parts, inventing a new component that relates to friction.

8.2.1.3.2 Discovery of Neutrons The predictive definition of atomic mass of some atom is proportional to the atomic number, i.e. the number of protons in an atom of an element. However, observations conflict with this prediction, because atomic mass is a measurement of the number of particles in an atom's nucleus. We can instantiate WMS with the following:

$$\{Pred/O_1, Obs/O_2, AtomicMass/stuff, \lambda x, y.x(y)/f, M/v\} \quad (8.56)$$

where *AtomicMass* is the function for calculating the mass of an atom and M is some measured value. The repair is to split *AtomicMass* into parts, creating an invisible notion of atomic mass which corresponds to the existence of neutrons.

8.2.1.3.3 Discovery of Neutrinos This example is similar to the discovery of neutrons, except it is at the subatomic level and the inference is more complex. The inference of the trigger formula comes from a disparity with the predicted value of total energy. However, the substitution is the same as (8.56).

8.2.1.4 Summary

We have successfully applied WMS to each of the two case studies, as described in depth, and other additional test examples. In all of the examples, the historically correct repair was within the search space, i.e. the invisible stuff created corresponds directly to the new concept that was conjectured in the historical episode. In the two examples reported extensively, the concept of elasticity energy was created for the bouncing-ball paradox and some hidden galaxy for the discovery of dark matter. This demonstrates that our methods cover both everyday modelling, e.g., bouncing-ball, and major conceptual advances, e.g., dark matter.

8.2.2 Inconstancy

The Inconstancy repair plan (§4.5, p.56) is triggered when there is a conflict between the predicted independence and the observed dependence of a function on some parameter, i.e., the observed value of a function unexpectedly varies when it is predicted to remain constant. This generally requires several observational theories, each with different observed values of the function, as opposed to the one observational theory in the WMS plan. To effect the repair, the parameter causing the unexpected variation is first identified and a new definition for the conflicting function is created that includes this new parameter.

8.2.2.1 Test Case Study I: The Travel Time of Light

One of the earliest recorded discussions of the speed of light, and thus the travel time of light, was by Aristotle, who believed that light travelled instantaneously and rejected theories about finite speeds of light. In 1676, a Danish astronomer, Ole Roemer, measured the speed by studying Io, one of Jupiter's moons, which was known to be eclipsed by Jupiter at regular intervals (Ellis and Uzan, 2005). Roemer discovered that the eclipses increasingly lagged behind the predicted times, but then started to pick up again. This discovery helped him develop the theory that when Jupiter and Earth were further apart, there was more distance for light reflecting off Io to travel to Earth and therefore it took longer to reach his telescope. Essentially Aristotle's theory that light appeared instantaneously can be repaired by Roemer's observations of variations in the occurrence times between eclipses of Io, as seen from Earth, to produce the correct theory that light has a finite speed. The required Physics background is relatively elementary, so the specific equations on which this case study depends are not presented here.

Suppose Aristotle's claim that the travel time of light is instantaneous is an assertion and the lag of eclipses Io measured by Roemer were recorded when the distances between the earth and Io was d_1 and d_2. In Figure 8.6, the axiomatisation formalises the representation of knowledge required to reason about the case study, where

- $(Dst{::}Point \Rightarrow Point \Rightarrow \mathbb{R})(p,q)$ returns the length of the straight-line path separating points p and q in a 3-dimensional space;

- $OrbPos(o,t)$ returns the orbital position of an object o at moment t;

$$Ax(Pred) \supseteq \{TravelTime(LightFrom(Io), Io, Earth) = 0, \tag{8.57}$$
$$ReactionTests(roemer) = \{test1, test2\}, \tag{8.58}$$
$$ReactionTime(roemer, test1) = 0.1, \tag{8.59}$$
$$ReactionTime(roemer, test2) = 0.2\} \tag{8.60}$$
$$Ax(Obs(Dst(Earth, \tag{8.61}$$
$$OrbPos(Io, T_1)) = d_1)) \supseteq \{NetTime(o, s, d, p) = \tag{8.62}$$
$$Approx(TravelTime(o, s, d)) + ReactionTime(p), \tag{8.63}$$
$$NetTime(LightFrom(Io), Io, Earth, roemer) = 10\}$$
$$Ax(Obs(Dst(Earth, \tag{8.64}$$
$$OrbPos(Io, T_2)) = d_2)) \supseteq \{NetTime(o, s, d, p) = \tag{8.65}$$
$$Approx(TravelTime(o, s, d)) + ReactionTime(p), \tag{8.66}$$
$$NetTime(LightFrom(Io), Io, Earth, roemer) = 15\}$$

where *Pred* contains assertions about Aristotle's belief that light travels instantaneously and reaction times of Roemer observed over two test sessions; and, $Obs(Dst(Earth, OrbPos(Io, T_1)) = d_1)$ and $Obs(Dst(Earth, OrbPos(Io, T_2)) = d_2)$ are observation ontologies containing data captured at moments T_1 and T_2, at which the distances between earth and the orbital position of Io are d_1 and d_2, respectively.

Figure 8.6: Summary of the axiomatisation of the travel time of light case study.

- $(TravelTime :: Obj \Rightarrow Point \Rightarrow Point \Rightarrow \mathbb{R})(o, s, d)$ returns the time an object o takes to travel from the point s to point d;
- $(Approx :: \mathbb{R} \Rightarrow \mathbb{R})(v)$ returns an approximated form of the value v;
- $(NetTime :: \mathbb{R} \Rightarrow \mathbb{R} \Rightarrow \mathbb{R})(t, r)$ returns the actual time recorded after taking into consideration the observer's response time and the object's traveltime. The argument t is the number of seconds the object takes to reach the observer and r is the reaction time of the observer in seconds;
- $(ReactionTime :: Person \Rightarrow \mathbb{R})(p)$ returns the least number of seconds a person p reacts to a stimulus
- $(ReactionTime :: Person \Rightarrow Event \Rightarrow \mathbb{R})(p, s)$ returns the number of seconds a person p responds to a stimulus as observed in a test session s;

- $(LightFrom::Obj \Rightarrow Obj)(o)$ returns the light object emitted from an object o;
- $roemer{:}Person$ represents the human observer who conducted the observation;
- $ReactionTests(roemer){:}Eventset$ is the set containing all the test sessions conducted to measure Roemer's reaction times; and,
- $test1{:}Event$ and $test2{:}Event$ represent two sessions for testing Roemer's reaction time.

The ontology *Pred* does not entirely share its vocabulary with the sensory ontologies, i.e. $Obs(Dst(Earth, OrbPos(Io, T_1)) = d_1)$ and $Obs(Dst(Earth, OrbPos(Io, T_2)) = d_2)$, e.g., *NetTime* is not in the language of *Pred*. Also, *ReactionTime* is a binary function in *Pred* but an unary function in the sensory ontologies. To reason with heterogeneous ontologies, we can identity an ontology in $\mathcal{F}(Pred)$ such that the merge between a sensory ontology, the bridging ontology, and this ontology does not induce inconsistency. Suppose this ontology is $Pred_f$:

$$Ax(Pred_f) \ ::= \ \{$$

$$ReactionTests(roemer) = \{test1, test2\}, \tag{8.67}$$

$$ReactionTime(roemer, test1) = 0.1, \tag{8.68}$$

$$ReactionTime(roemer, test2) = 0.2\} \tag{8.69}$$

$$\}$$

As the axioms (8.67 - 8.69) make a subset of those of *Pred*, $Pred_f$ is guaranteed to be a node in $\mathcal{F}(Pred)$.

To relate the *ReactionTime* symbol in the predictive ontology to that in the sensory ontologies, we encode the following bridging axiom in O_b:

$$\begin{aligned} &\forall p : Person,\ v : \mathbb{R}. \\ &\quad Obs(Dst(Earth, \\ &\quad OrbPos(Io, T_1)) = d_1) \vdash ReactionTime(p) = v \longleftrightarrow \\ &\quad Pred_f \vdash \min_{t \in ReactionTests(p)} ReactionTime(p, t) = v \end{aligned} \tag{8.70}$$

$$\begin{aligned} &\forall p : Person,\ v : \mathbb{R}. \\ &\quad Obs(Dst(Earth, \\ &\quad OrbPos(Io, T_2)) = d_2) \vdash ReactionTime(p) = v \longleftrightarrow \\ &\quad Pred_f \vdash \min_{t \in ReactionTests(p)} ReactionTime(p, t) = v \end{aligned} \tag{8.71}$$

where (8.70) and (8.71) relate the notion of binary version of reaction time in *Pred* to the unary version of reaction time in *Obs* by stating that the value of a person's reaction time in *Obs* is equal to the minimum value of that person's reaction times measured over a number of tests in *Pred*.

From these ontologies, we can formulate a contradiction as follows:

$$Pred \vdash TravelTime(LightFrom(Io), Io, Earth) := 0 \tag{8.72}$$

$$O_b \oplus Pred_f \oplus Obs(Dst(Earth, OrbPos(Io, T_1)) = d_1) \vdash Approx(TravelTime(LightFrom(Io), Io, Earth)) = 9.9$$

$$O_b \oplus Pred_f \oplus Obs(Dst(Earth, OrbPos(Io, T_2)) = d_2) \vdash Approx(TravelTime(LightFrom(Io), Io, Earth)) = 14.9$$

$$Pred \vdash 9.9 \neq 14.9 \tag{8.73}$$

The detected conflict can trigger the Inconstancy repair plan with the following substitution:

$$\begin{aligned}&\{Pred/O_x, Obs/O_y, Pred_f \oplus O_b/O_z,\\ &Dst/v, \lambda x.x(Earth, OrbPos(Io, T_1))/b_1,\\ &\lambda x.x(Earth, OrbPos(Io, T_2))/b_2, d_1/v_1, d_2/v_2,\\ &TravelTime(LightFrom(Io), Io, Earth)/stuff,\\ &Approx/f, 0/c, 9.9/c_1, 14.9/c_2\}\end{aligned} \tag{8.74}$$

Because we perform matching with $f(stuff)$, $stuff$ can be instantiated to

$$TravelTime(LightFrom(Io), Io, Earth) \tag{8.75}$$

which is the argument of *Approx*.

To effect the repair, we first insert the variad, $Dst(Earth, OrbPos(Io, T_i))$, into the new definition by following (4.49, p.60) to get:

$$\nu(TravelTime(LightFrom(Io), Io, Earth) ::= \lambda y.\ F(0, y(Dst)) \tag{8.76}$$

which means that the travel time of the light emitted by Io to travel to Earth now depends on the distance of some path, which is exactly what is required because light has a finite speed. The value the argument of *Dst* takes depends on the ontology. The repaired ontologies are shown in Figure 8.7.

$$Ax(\nu(Pred)) \supseteq \{\nu(TravelTime(LightFrom(Io), Io, Earth) = \lambda y.\ F(0, Dst(y))\} \tag{8.77}$$

$$Ax(\nu(Obs(Dst(Earth, OrbPos(Io, T_1)) = d_1))) \supseteq \{Approx(\nu(TravelTime(LightFrom(Io), Io, Earth)) (\lambda x.x(Earth, OrbPos(Io, T_1)))) = 10\} \tag{8.78}$$

$$Ax(\nu(Obs(Dst(Earth, OrbPos(Io, T_2)) = d_2))) \supseteq \{Approx(\nu(TravelTime(LightFrom(Io), Io, Earth)) (\lambda x.x(Earth, OrbPos(Io, T_2)))) = 15\} \tag{8.79}$$

where $\nu(Pred)$ contains the new definition of the travel time of light emitted from Io, which depends on the distance of some path and each of the new observation ontologies has the relevant argument applied to *stuff*.

Figure 8.7: Summary of the repaired axiomatisation of the travel time of light case study

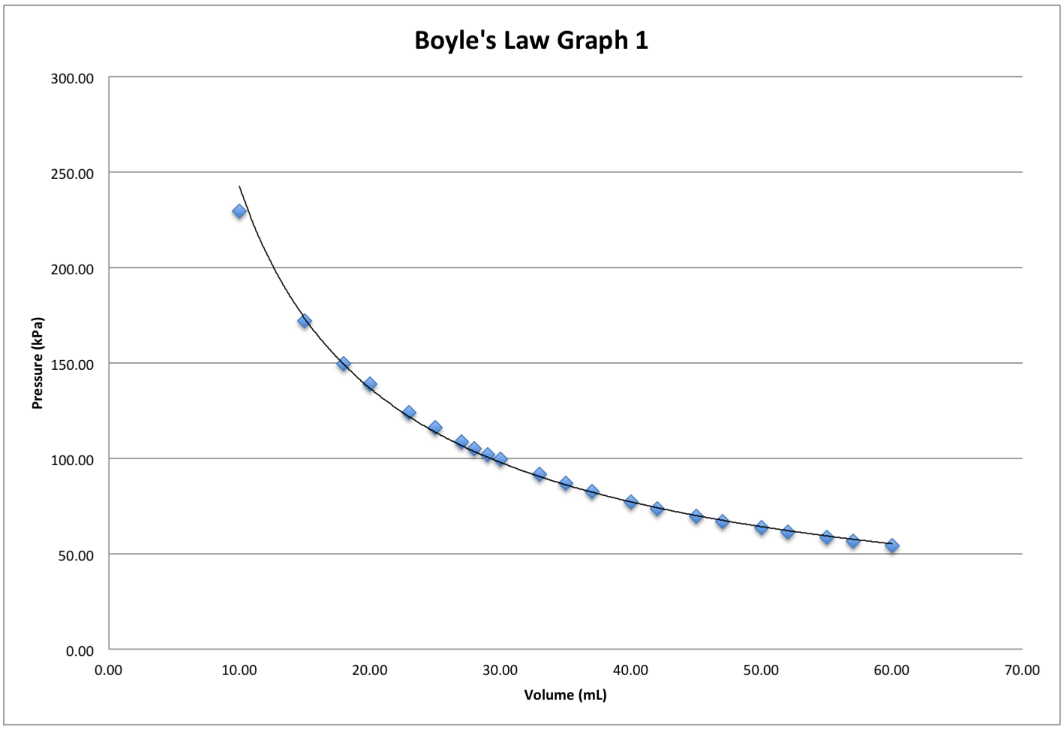
8.2.2.2 Test Case Study II: Gas Laws

An example from introductory Physics is Boyle's Law, which formulates a relationship between the amount of pressure and the volume of a gas. The law was discovered by Robert Boyle in 1662 and it states that given a fixed amount of gas (molecules) n at a fixed temperature T, the pressure P and the volume V of the gas are inversely proportional to each other, i.e.

$$P \times V = k \tag{8.80}$$

where k is a constant. Figure 8.8 illustrates a graph of Boyle's original data.

Boyle's Law is most famous for being the basis of derivation for the ideal gas law, which provides a complete formulation of the relationship between the pressure, the volume, the temperature, and the amount of gas. In this example, we will describe how the Inconstancy plan can be applied to modify Boyle's Law to resemble the ideal gas law.



This diagram is taken from `http://en.wikipedia.org/wiki/Boyle's_law`.

Figure 8.8: A graph of pressure-volume based on Boyle's original data, where the x-axis is the volume of the mercury vapour used in an experiment and the y-axis is the amount of pressure exerted.

Similar to the example in §8.2.2.1, we want to identify inconsistencies between the theoretical and observed variable dependences. It is important to note that Boyle's Law is regarded to be a correct account of the relationship between pressure and volume at a fixed temperature and amount of gas in modern science, but the law itself is simply incomplete. To fit our purposes, we alter the historic scenario slightly and introduce an inconsistency. We can model the scenario by letting *Pred* be the ontology modelling Boyle's Law and that the value of the product of pressure and volume is the same at *any* temperature. Suppose $Boyle(g)$ returns the graph of the product of pressure and volume of a gas g versus time, which is expected to be a constant graph. For mercury, the y-value is expected to be always 1,400, based on readings in Figure 8.8. Let $Obs(Temp(g,T_i) = v_i)$ be the sensory ontology describing the situation when the amount of the gas g is fixed and the temperature at a time moment T_i is v_i. We need to collect evidence for different temperatures over a range of time moments: T_i for

$1 \leq i \leq n$, where $Temp(g, T_i)$ varies.

The axiomatisation of the gas laws case study is summarised in Figure 8.9, where:

- *Gas* is the type of gas
- $(Boyle :: Gas \Rightarrow Mom \Rightarrow \langle Mom, \mathbb{R} \rangle)(g)$ returns a graph plotting time against the product of the pressure and volume of the gas g;
- $(Pres :: Gas \Rightarrow Mom \Rightarrow \mathbb{R})(g, t)$ returns the amount of pressure exerted on a gas g at a time moment t;
- $(Vol :: Gas \Rightarrow Mom \Rightarrow \mathbb{R})(g, t)$ returns the volume of a gas g at a time moment t;
- $(Hg :: Obj \Rightarrow Mod \Rightarrow Obj)(o, m)$ returns the mercury content with respect to an object o, according to the modifier m; if m is of value *In*, then it returns the mercury content *inside o*;
- $(Flask :: Obj \Rightarrow Mod \Rightarrow Obj)(o, m)$ returns the flask with respect to object o, according to the modifier m; if m is of value *On*, then it returns the flask placed *on o*;
- *Desk* is a desk object in the laboratory;
- $(T_1 Graph :: Gas \Rightarrow Mom \Rightarrow \langle Mom, \mathbb{R} \rangle)(g)$ returns a graph object plotting Boyle's values based on observations made at moment T_1;
- $(T_2 Graph :: Gas \Rightarrow Mom \Rightarrow \langle Mom, \mathbb{R} \rangle)(g)$ returns a graph object plotting Boyle's values based on observations made at moment T_2;
- $(Height :: Obj \Rightarrow Mom \Rightarrow \mathbb{R})(o, t)$ returns the vertical height of an object o at a time moment t; and,
- $(Surface :: Obj \Rightarrow Mom \Rightarrow \mathbb{R})(o, t)$ returns the area of a surface on an object o at a time moment t.

In the axiomatisation, the product of the pressure and volume of mercury vapour is asserted to be 1,400 under all circumstances. The experiments conducted are presumed to involve measuring both the surface area of the top of the flask used to contain the vapour, which is placed on a desk, and the height of the flask relative to its bottom. In each observation, the height of the flask and the surface area of its top were observed to be the same. Unexpectedly, the pressure of the vapour content changed over the two

$$Ax(Pred) \supseteq \{\forall g{:}Gas.\ Boyle(g) = \tag{8.81}$$

$$\lambda t.\ \langle t, Pres(g,t) \times Vol(g,t)\rangle,$$

$$\forall t{:}Mom.\ Pres(Hg(Flask(Desk,On),In),t) \times \tag{8.82}$$

$$Vol(Hg(Flask(Desk,On),In),t) = 1400$$

$$\}$$

$$Ax(Obs(Temp(Hg(Flask(Desk,$$

$$On),In),T_1) = 20) \supseteq \{\forall g{:}Gas, t{:}Mom.\ T_1Graph(g,t) = \tag{8.83}$$

$$Pres(g,t) \times Height(Flask(Desk,On),t) \times$$

$$Surface(Flask(Desk,On),t),$$

$$Pres(Hg(Flask(Desk,On),In),T_1) = 80, \tag{8.84}$$

$$Surface(Flask(Desk,On),T_1) = 2, \tag{8.85}$$

$$Height(Flask(Desk,On),T_1) = 10\} \tag{8.86}$$

$$Ax(Obs(Temp(Hg(Flask(Desk, \tag{8.87}$$

$$On),In),T_2) = 25) \supseteq \{\forall g{:}Gas, t{:}Mom.\ T_2Graph(g,t) = \tag{8.88}$$

$$Pres(g,t) \times Height(Flask(Desk,On),t) \times$$

$$Surface(Flask(Desk,On),t),$$

$$Pres(Hg(Flask(Desk,On),In),T_2) = 100, \tag{8.89}$$

$$Surface(Flask(Desk,On),T_2) = 2 \tag{8.90}$$

$$Height(Flask(Desk,On),T_2) = 10\} \tag{8.91}$$

where *Pred* contains the predictive theory based on Boyle's Law and $Obs(Temp(Hg(\ldots)) = 20)$ and $Obs(Temp(Hg(\ldots)) = 25)$ are observation ontologies containing data captured at two different moments, at which the temperatures of the mercury vapour concerned were 20 degrees and 25 degrees, respectively.

Figure 8.9: Summary of the axiomatisation of the gas laws case study.

observations: 80 when the temperature was 20 and 100 when the temperature was 25. Thus, the products of the pressure and volume of the vapour were 1,600 and 2,000, respectively.

The ontology *Pred* does not entirely share its vocabulary with the sensory ontologies, e.g., *Boyle* is not in the language of the sensory ontologies. Similar to previous case studies, we can identity an ontology in $\mathcal{F}(Pred)$ such that the merge between a sensory ontology, the bridging ontology, and this ontology does not induce inconsistency. Suppose this ontology is $Pred_f$:

$$Ax(Pred_f) \quad ::= \quad \{\, \forall g{:}Gas.\ Boyle(g) = \lambda t.\ \langle t, Pres(g,t) \times Vol(g,t)\rangle \,\} \tag{8.92}$$

As the axiom (8.92) is within of those of *Pred*, $Pred_f$ is guaranteed to be a node in $\mathcal{F}(Pred)$.

To relate the symbol *Boyle* in the predictive ontology to symbols in the sensory ontologies, we encode the following bridging axioms in O_b:

$$\begin{aligned} &\forall g : Gas,\ v : Mom \Rightarrow \langle Mom, \mathbb{R}\rangle. \\ &\quad Obs(Temp(Hg(Flask(Desk, \\ &\quad On), In), T_1) = 20)\ \vdash\ \lambda t.\langle t, T_1 Graph(g,t)\rangle = v \longleftrightarrow \\ &\quad Pred_f\ \vdash\ Boyle(g) = v \end{aligned} \tag{8.93}$$

$$\begin{aligned} &\forall g : Gas,\ v : Mom \Rightarrow \langle Mom, \mathbb{R}\rangle. \\ &\quad Obs(Temp(Hg(Flask(Desk, \\ &\quad On), In), T_2) = 25)\ \vdash\ \lambda t.\langle t, T_2 Graph(g,t)\rangle = v \longleftrightarrow \\ &\quad Pred_f\ \vdash\ Boyle(g) = v \end{aligned} \tag{8.94}$$

where (8.93) and (8.94) relate the notion of Boyle's graph in *Pred* to symbols in *Obs*.

From these ontologies, we can formulate a contradiction as follows:

$$Pred \vdash Boyle(Hg(Flask(Desk,On),In)) = \tag{8.95}$$

$$\lambda t.\ \langle t, 1400\rangle \tag{8.96}$$

$$O_b \oplus Pred_f \oplus Obs(Temp(Hg(Flask(Desk,On),In),T_1) = 20) \vdash Boyle(Hg(Flask(Desk,On),In)) = \tag{8.97}$$

$$\lambda t.\ \langle t, T_1 Graph(Hg(Flask(Desk,On),In),t)\rangle$$

$$O_b \oplus Pred_f \oplus Obs(Temp(Hg(Flask(Desk,On),In),T_2) = 25) \vdash Boyle(Hg(Flask(Desk,On),In)) = \tag{8.98}$$

$$\lambda t.\ \langle t, T_2 Graph(Hg(Flask(Desk,On),In),t)\rangle$$

$$Pred \vdash \lambda t.\ \langle t, 1400\rangle \neq \tag{8.99}$$

$$\lambda t.\ T_1 Graph(Hg(Flask(Desk,On),In),t)$$

In each of the three ontologies, we can infer a graph object to represent values returned by Boyle's law for the mercury vapour over time. Here, we know at least two of the graph objects are regarded to be different (8.99). Thanks to the polymorphic $\neq$ operator, we are allowed to directly compare two graph objects. We know the graph $\lambda t.\ \langle t, 1400\rangle$ must have the following property

$$\forall s.\ (\lambda t.\ snd(\langle t, 1400\rangle)(s)) = 1400 \tag{8.100}$$

where $snd(p)$ returns the second value of the pair p. We can infer from (8.83), (8.84), (8.85), and (8.86) that

$$snd(\langle T_1, T_1 Graph(Hg(Flask(Desk,On),In),T_1)\rangle) = 1600 \tag{8.101}$$

With (8.100) and (8.101), (8.99) can be resolved.

We can match the trigger formulae (4.46 - 4.48) of Inconstancy with the following substitution:

$$\begin{aligned}&\{Pred/O_x, Obs/O_y, O_b \oplus Pred_f/O_z, Temp/v,\\&\lambda x.x(Hg(Flask(Desk,On),In),T_1)/b_1,\\&\lambda x.x(Hg(Flask(Desk,On),In),T_2)/b_2, 20/v_1, 25/v_2,\\&Boyle/stuff,\ \lambda x.\ x(Hg(Flask(Desk,On),In))/f,\\&\lambda t.\ \langle t,1400\rangle/c, \lambda t.T_1Graph(Hg(Flask(Desk,On),In),t)/c_1,\\&\lambda t.T_2Graph(Hg(Flask(Desk,On),In),t)/c_2\}\end{aligned} \tag{8.102}$$

Unlike the previous case study, variables c, c_1, and c_2 are all instantiated to graph objects here and *stuff* is instantiated to a dominant function.

To effect the repair, we create the following new definition:

$$\nu(Boyle) ::= \lambda y.\ F(\lambda t.\ \langle t,1400\rangle, y(Temp)) \tag{8.103}$$

which means that the value returned by the product of pressure and volume of a gas now depends on the temperature. The repaired ontologies are shown in Figure 8.10.

As a result of the repair, the repaired Boyle's Law is given the temperature of the gas as an additional argument, which drives the variable dependence of the repaired law closer to that required by the ideal gas law. Further, a very similar repair can be made upon the confrontation of evidence for varying amounts of gas n. The resulting list of dependent variables will subsequently include n, and so the further repaired Boyle's Law will depend on all four variables, as required by the ideal gas law.

8.2.2.3 Other Test Case Studies

In addition to the case studies covered in (§8.2.2.1) and (§8.2.2.2), we have also evaluated Inconstancy over a diverse range of other case studies. These include:

- *Variations in the Fine-Structure Constant*: The fine-structure constant is a number that determines the strength of interactions between light and matter. It is a fundamental physical constant, but some physicists believe its value varies over time.

- *Quasar Alignment Patterns*: The orientations of quasars in the outer universe are expected to be random, yet many of these point in a similar direction. Some physicists have discovered that the axis of rotation of the quasar depends on the

$$Ax(\nu(Pred_f)) \supseteq \{\nu(Boyle) = \lambda y.\, F(\lambda t.\, \langle t, 1400\rangle, y(Temp)), \quad (8.104)$$

$$\nu(Boyle)(y)(g) = \lambda t.\langle t, Pres(g,t) \times Vol(g,t)\rangle\} \quad (8.105)$$

$$\forall g : Gas,\ v : Mom \Rightarrow \langle Mom, \mathbb{R}\rangle. \quad (8.106)$$
$$Obs(Temp(Hg(Flask(Desk,$$
$$On), In), T_1) = 20) \vdash \lambda t.\langle t, T_1Graph(g,t)\rangle = v \longleftrightarrow$$
$$\nu(Pred_f) \vdash \nu(Boyle)(y)(g) = v$$

$$\forall g : Gas,\ v : Mom \Rightarrow \langle Mom, \mathbb{R}\rangle. \quad (8.107)$$
$$Obs(Temp(Hg(Flask(Desk,$$
$$On), In), T_2) = 25) \vdash \lambda t.\langle t, T_2Graph(g,t)\rangle = v \longleftrightarrow$$
$$\nu(Pred_f) \vdash \nu(Boyle)(y)(g) = v$$

where $\nu(Pred)$ contains the new definition of Boyle's Law, which depends on the temperature of the gas and each of the observation ontologies has the relevant argument applied to *stuff*; $\nu(Pred_f)$ applies $\nu(Boyle)$ to a variable y, as $Pred_f$ is part of the instantiation of O_z; and, the new bridging axioms speak about $\nu(Pred_f)$ instead of $Pred_f$ with previous occurrences of *Boyle* replaced by $\nu(Boyle)(y)$, as O'_b is part of the instantiation of O_z as well.

Figure 8.10: Summary of the repaired axiomatisation of the gas laws case study.

magnetic fields caused by cosmic strings resulted within 10 seconds after the big-bang.

- *Rate of Evolvability*: The rate at which species evolve was thought to be constant, but some postulate that the rate may depend on *mutational robustness*, which is the capacity to develop normally despite the presence of genetic mutations.

8.2.2.3.1 The Changing Fine-Structure Constant Suppose the experiment involves making two or more observations of distant galaxies and the fine-structure constant is derived from these observations. We can match the trigger formulae of Inconstancy with the following substitution[8]:

$$\{Pred/O_x, Obs/O_y, TimeOf/v, obs_1/b_1, obs_2/b_2, \quad (8.108)$$
$$FSC/stuff,\ \lambda x.x/f, K/c, K_1/c_1, K_2/c_2\}$$

[8] We here simplify by excluding O_z from the discussion.

where $TimeOf(obs)$ returns the time when the observation obs was conducted; obs_1 and obs_2 are two different observation events; FSC is the fine-structure constant; and, K, K_1, and K_2 are distinct constants.

8.2.2.3.2 Quasar Alignment Patterns In the formalisation of this case study, the axis of rotation of a particular quasar is expected to be constant, but observations show that it actually varies with the density of cosmic strings. Suppose the axis of rotation of a particular quasar can be measured seconds after the big bang[9]. We can instantiate the trigger formulae of Inconstancy (4.46 - 4.48) as follows[10]:

$$\{Pred/O_x, Obs/O_y, CosmicStrings/v, \lambda x.x(Q,T_1)/b_1, \quad (8.109)$$
$$\lambda x.x(Q,T_2)/b_2, RotAxis/stuff,\ \lambda x.x(Q)/f, A/c, A_1/c_1, A_2/c_2\}$$

where $Pred$ and Obs are the predictive and observation ontologies, respectively; $CosmicStrings(q)$ returns the density of cosmic strings near the region of the quasar q; Q is the quasar under consideration; $RotAxis(q)$ returns the axis of rotation of the quasar q; and A, A_1, and A_2 are vectors representing three different axes of rotation.

8.2.2.3.3 Rate of Evolvability Our model of the case study is based an unexpected variation in the rate of evolvability of a species as its mutation robustness changes. Inconstancy can be triggered by the following substitution[11]:

$$\{Pred/O_x, Obs/O_y, MutRob/v, species_1/b_1, species_2/b_2, \quad (8.110)$$
$$RateEvolve/stuff,\ \lambda x.x/f, K/c, K_1/c_1, K_2/c_2\}$$

where $Pred$ and Obs are the predictive and observation ontologies, respectively; $MutRob(s)$ is the function returning the capacity in terms of mutation robustness of a species s; $species_1$ and $species_2$ are two different species exhibiting different amounts of mutation robustness; $RateEvolve$ is the rate of evolvability, which is expected to be constant; K, K_1, and K_2 are distinct constants.

8.2.2.4 Summary

The Inconstancy repair plan has produced meaningful repairs to each of the two case studies described that eliminated the detected contradiction. We have demonstrated

[9] This should be a plausible task, given that astronomers are observing the state of the distant universe moments after the big bang.

[10] We here simplify by excluding O_z from the discussion.

[11] We here simplify by excluding O_z from the discussion.

how the repairs performed by this repair plan transform inconsistent ontologies into new theories that closely match true physical formulations. The use of heterogeneous ontologies has somewhat increased the complexity of the execution, e.g., the variable O_z also needs to be instantiated. However, we have shown that diagnosing and repairing faults with Inconstancy can intuitively lead to a resolution.

8.2.3 Unite

We explore the Unite ontology repair plan (§4.6, p.62), which can be seen as the converse of WMS. If two terms yield the same value for their defining property, then they should refer to the same thing. The idea of Unite is to take two different terms and equate them, provided that their defining property yields the same value.

8.2.3.1 Test Case Study I: Quantisation of Space-Time

Planck units define the smallest discrete units of space, time, mass and energy. The values of these units are all very small; for instance, the Planck length is approximately a 10 billion billionth of the width of a proton. Since there is a minimum interval of time (Planck time) as well, or a maximum frequency in nature, there is a corresponding limit on the fidelity of space and time. An analogous situation is having noise in an audio stream, which is a result of adopting a low resolution/pixelation in the recording or transmission. To detect a similar kind of noise in nature, a team at Fermilab are constructing an apparatus called a *holometer*, which will fire two laser beams from a single source and they will measure the resulting jitter (holographic noise) as the two beams are recombined (FermiLab, 2012). If there is pixelation in nature, then the two beams will not be travelling in the same direction or at the same time, which means they will not probe the same volume of space-time. In that case, the measurements will have extra, uncorrelated jitter. Their conjecture is that the smallest units are in fact larger than the Planck units, as suggested by some recent theoretical studies of black holes. In this case study, we suppose the empirical experiment confirms the values of Planck units and that the holometer is, at least, not sensitive to detect uncorrelated jitters when the two beams are recombined.

We can use the Unite repair plan to emulate this episode as follows. Suppose the predictive ontology, *Pred*, does not speak about the relationship between the volume of space-time probed by the two beams, *BeamA* and *BeamB*, but only defines the meaning

$$Ax(Pred) \supseteq \{\forall k_1, k_2{:}JitterKind.\ Correlate(k_1, k_2) \longleftrightarrow \quad (8.111)$$
$$|\max_x k_1(x) - \max_x k_2(x)| \leq 2\}$$
$$Ax(Obs) \supseteq \{Jitter(BeamA) = KindA, \quad (8.112)$$
$$Jitter(BeamB) = KindB, \quad (8.113)$$
$$\forall x{:}\mathbb{R}.\ 0 < KindA(x) < 1, \quad (8.114)$$
$$\forall x{:}\mathbb{R}.\ 1 < KindB(x) < 2\} \quad (8.115)$$
$$Ax(O_M) \supseteq \{DefProp(Beam, Jitter)\} \quad (8.116)$$

where *Pred* contains the predictive theory, in which the meaning of two kinds of jitter being correlated is asserted; *Obs* is the observation ontology containing measurements of the jitter resulting from recombining two light beams; and, O_m is the meta-level ontology containing the definition of the relevant defining property.

Figure 8.11: Summary of the axiomatisation of the quantisation of space-time case study.

of two kinds of jitters being correlated and the observation ontology, *Obs*, specifies the values of the points plotted in the jitter-graph.

An axiomatisation of the case study is summarised in Figure 8.11, where

- $(Correlate :: JitterKind \Rightarrow JitterKind \Rightarrow bool)(k_1, k_2)$ returns true if and only if two kinds of jitters k_1 and k_2 are deemed correlated, i.e. if the difference between the maximum y-values of k_1 and k_2 is less than or equal to two[12];
- $(Jitter :: Beam \Rightarrow JitterKind)(b)$ returns the kind of jitter corresponding to a light beam b; and,
- $(SpaceTimeVol :: Beam \Rightarrow STVol)(b)$ returns the volume of space-time probed by the light beam b.

We assume that the jitter corresponding to a light beam can be classified into a particular *kind*, which is essentially a graph object. To compute the correlation between two kinds of jitters, we have adopted a simple notion: if two kinds of jitters are correlated, then the difference between their maximum y-values must be less than or equal to two (8.111). In the observation, the kinds of jitters corresponding to light beams *BeamA*

[12]We use the value two purely for explanatory purposes.

and *BeamB* are *KindA* and *KindB*, respectively (8.112, 8.113). By definition, all y-values of *KindA* are within the interval (0,1) (8.114), whereas all y-values of *KindB* are within (1,2) (8.115). The defining property of the type *Beam* is specified in the meta-ontology O_M (8.116). The *Obs* ontology is not an extension of *Pred*, so the symbol *Correlate* is not in its language. Unlike previous case studies, Unite deals with under-specified ontologies, so the merge of the input ontologies is still consistent. Thus, we can resolve the heterogeneity in the languages of *Pred* and *Obs* by simply reasoning with $Pred \oplus Obs$. Nonetheless, a bridging axiom can still be helpful:

$$\forall k_1, k_2{:}JitterKind.\ Pred \vdash Correlate(k_1,k_2) = True \longleftrightarrow Obs \vdash k_1 = k_2 \qquad (8.117)$$

To keep the formalisation simple, we have indirectly overloaded $=_{JitterKind}$ in *Obs* using (8.117)[13].

From our formalisation, we can show the following:

$$\begin{aligned} Pred \ &\nvdash\ SpaceTimeVol(BeamA) = SpaceTimeVol(BeamB) && (8.118)\\ Obs \oplus O_b \oplus & \\ Pred \ &\vdash\ Jitter(BeamA) = Jitter(BeamB) && (8.119)\\ O_M \ &\vdash\ DefProp(Beam, Jitter). && (8.120) \end{aligned}$$

We can instantiate the trigger formulae of the Unite repair plan with the following substitution:

$$\{Pred/O_1, Obs \oplus O_b \oplus Pred/O_2, Beam/\tau, BeamA/stuff_1,$$
$$BeamB/stuff_2, Jitter/dp, SpaceTimeVol/f\}$$

Following (4.66, p.64) of the repair plan, the corresponding repair is then:

$$Ax(\nu(Pred)) \ ::= \ \{SpaceTimeVol(BeamA) = SpaceTimeVol(BeamB)\} \cup Ax(Pred)$$

which means *BeamA* and *BeamB* are asserted to probe the same volume of space-time. This is exactly the experimentalists' expectation should there be a correlation between the corresponding jitters. If such a result is confirmed by the experiment, then the validity of Planck's units hold.

[13]The overloading of $=_{JitterKind}$ could instead be specified in the axioms of *Pred* or *Obs*.

8.2.3.2 Test Case Study II: The Bouncing-ball Paradox Revisited

We can demonstrate the power of the Unite repair plan by applying it to the Bouncing-ball Paradox (8.2.1.1) as well. Suppose the language of the formalisation is slightly extended by the addition of four object-level symbols:

- $ClassOfObj(o)$ is designed for evaluating the class of objects to which an object o belongs. The values returned by $ClassOfObj(o)$ could represent elastic solids, particles without extent, liquids and gases. The specific values are not of particular importance in this example. Note that classes are encoded as types in 8.2.1.1 instead. With an object-level symbol, we can reason about object classes more naturally.
- A predicate $IsElastic(o)$ returns true if and only if o exhibits elasticity
- A predicate $IsGaseous(o)$ returns true if and only if o is a gas
- A predicate $IsLiquidy(o)$ returns true if and only if o is a liquid.

Same as in 8.2.1.1, the object concerned is an elastic, bouncing-ball, *Ball*.

Suppose the predictive theory here is an extension of that in 8.1 and the experiment was conducted in the same manner, i.e. by taking a series of photos of the ball separated by a positive constant interval $\delta{:}\mathbb{R}$ while it is dropped.

Figure 8.12 shows an extended axiomatisation of the bouncing-ball paradox, where (8.121) states that the sum of kinetic and potential energies is not conserved for elastic objects; (8.122) asserts that *Ball* is neither a gas nor liquid; and, (8.123) asserts that a spring *Spring* is neither a gas nor liquid but is elastic. In O_M, the defining property for the type *Obj* w.r.t. object classes is asserted to depend on the state of the object and its elasticity. More specifically, if two objects are neither gases nor liquids but are elastic, then they belong to the same object class.

Along with the bridging axiom in (8.21), we can show the following:

$$Pred \nvdash ClassOfObj(Ball) = ClassOfObj(Spring) \tag{8.135}$$

$$Obs \oplus Pred \vdash IsElastic(Ball) \wedge \neg IsGaseous(Ball) \wedge \tag{8.136}$$

$$\neg IsLiquidy(Ball) = IsElastic(Spring) \wedge \tag{8.137}$$

$$\neg IsGaseous(Spring) \wedge \neg IsLiquidy(Spring) \tag{8.138}$$

$$O_M \vdash DefProp(Obj, \lambda x.IsElastic(x) \wedge \neg IsGaseous(x) \wedge \tag{8.139}$$

$$\neg IsLiquidy(x)) \tag{8.140}$$

$$
\begin{aligned}
Ax(Pred) \supseteq \{ & \\
& \forall o{:}Obj,\ t_1,t_2{:}Mom.\ KE(o,t_1)+PE(o,t_1) \neq && (8.121) \\
& KE(o,t_2)+PE(o,t_2) \longleftrightarrow IsElastic(o), \\
& \neg IsGaseous(Ball) \wedge \neg IsLiquidy(Ball), && (8.122) \\
& \neg IsGaseous(Spring) \wedge \neg IsLiquidy(Spring) \wedge && (8.123) \\
& IsElastic(Spring), \\
& \forall o{:}Obj,t{:}Mom.KE(o,t) = \frac{1}{2}Mass(o,t) \times Velocity(o,t)^2, && (8.124) \\
& \forall o{:}Obj,t{:}Mom.PE(o,t) = Mass(o,t) \times G \times Height(o,t), && (8.125) \\
& Mass(Ball,Start(Drop)) > 0, && (8.126) \\
& G > 0, && (8.127) \\
& Height(Ball,Start(Drop)) > 0, && (8.128) \\
& Velocity(Ball,Start(Drop)) = 0 && (8.129) \\
& \} \\
Ax(Obs) \supseteq \{ & \\
& Height(Ball,End(Drop)) = 0, && (8.130) \\
& Pos(Ball,Photo(End(Drop)-1)) = (0,0), && (8.131) \\
& Pos(Ball,Photo(End(Drop))) = (0,0) && (8.132) \\
& \} \\
Ax(O_M) \supseteq \{ & DefProp(Obj,\lambda x.IsElastic(x) \wedge \neg IsGaseous(x) \wedge \neg IsLiquidy(x) && (8.133) \\
& \wedge, ClassOfObj)\} && (8.134)
\end{aligned}
$$

where both *Pred* and *Obs* here are the same as those in (8.1) except for the addition of (8.121), (8.122) and (8.123) in *Pred*. Note that axioms containing *TE* are omitted due to their irrelevance to this case study.

Figure 8.12: Summary of an axiomatisation of the revisit of bouncing-ball paradox.

Thus, the trigger formulae of the Unite repair plan can be instantiated as the following:

$$\{Pred/O_1, Obs \oplus Pred/O_2, Obj/\tau, Ball/stuff_1, Spring/stuff_2,$$
$$\lambda x.IsElastic(x) \wedge \neg IsGaseous(x) \wedge \neg IsLiquidy(x)/dp, ClassOfObj/u\}$$

and effect the repair by inserting a new axiom:

$$Ax(\nu(Pred)) \quad ::= \quad \{ClassOfObj(Ball) = ClassOfObj(Spring)\} \cup Ax(Pred)$$

which means *Ball* and *Spring* now belong to the same object class as required. Both *Ball* and *Spring* are in fact elastic materials, so it is expected that they are classified under the same object class.

8.2.3.3 Summary

The notion of defining property is at the crux of the Unite repair plan. Defining properties of physical concepts can be naturally derived partly because of the formal approach to understanding Physics adopted by the community. Unite tackles the problem of under-specification, so no contradiction arises even if the ontologies in question are merged. Its repair operation is relatively straightforward, but effectively increases the logical strength of the given ontologies.

8.2.4 Reidealisation

The Reidealisation repair plan (§4.4, p.51) repairs a kind of fault WMS detects and repairs. It resolves a special case of this kind of conflict without inventing some invisible component and taking the viewpoint that the original conceptualisation gives only a partial view of the underlying property. Instead, the idealisation of the property could be changed such that it is viewed as being a property of another type. With the new idealisation, the measurement function, which originally returns different values in each of the two ontologies, will no longer return conflicting values.

8.2.4.1 Test Case Study I: Demotion of Pluto

Before the discovery of Pluto, the existence of a ninth planet in the Solar System was long envisioned by astronomers, which they called Planet X. In 1930, Clyde Tombaugh declared that he had discovered Planet X and his team named it Pluto, which is a member of a region in the Solar System called the *Kuiper Belt*. Over the past few decades,

$$Sig(Pred) \supseteq \{Pluto{:}Planet\} \tag{8.141}$$

$$Ax(Pred) \supseteq \{ClearNeighbour(x) = \begin{cases} True & \text{if } x{:}Planet \\ False & \text{if } x{:}DPlanet \end{cases} \tag{8.142}$$

$$\forall x.ClearNeighbour(x) \longleftrightarrow \tag{8.143}$$

$$\forall x \neq y, t_1, t_2.Orbit(x,t_1) \neq Orbit(y,t_2)$$

$$Sig(Obs) \supseteq \{Pluto{:}Planet\} \tag{8.144}$$

$$Ax(Obs) \supseteq \{KuiperBelt = \{PlutinoA, PlutinoB, \ldots\}, \tag{8.145}$$

$$FromChart(PlutinoA, T_1) = CoordA, \tag{8.146}$$

$$FromChart(PlutinoB, T_1) = CoordB, \tag{8.147}$$

$$FromChart(Pluto, T_2) = CoordA, \tag{8.148}$$

$$Pluto \neq PlutinoA \neq PlutinoB \tag{8.149}$$

$$\} \tag{8.150}$$

where *Pred* contains the essential definitions for specifying the new classification and the notion of a clear neighbourhood, and *Obs* is the observation ontology containing measurements of the orbit of Pluto and the celestial objects positioned at various coordinates.

Figure 8.13: Summary of the axiomatisation of the demotion of Pluto case study.

the understanding of the outer Solar System has been dramatically changed by more powerful observatory technologies and new theoretical evidence. Astronomers have discovered increasingly larger objects in the Kuiper Belt, some of which are almost of the size of Pluto. In 2005, an object believed to be larger than Pluto was discovered, which prompted a review of the classification of Pluto and the general definition of a planet. After rigorous debates, astronomers voted for the controversial decision of demoting Pluto down from the classification of a Planet to a new classification of *dwarf planet*. For an object to qualify for the Planet classification, it needs to meet three requirements: a) it is in orbit around the Sun, b) it has sufficient mass for it to have enough self-gravity to pull itself into a nearly round shape, and c) has cleared the neighbourhood around its orbit. Pluto satisfies a) and b) but fails c). There are over 70,000 icy objects with the same composition of Pluto in the Kuiper Belt, so Pluto's orbit cannot be considered as being cleared We emulate the reasoning behind the demotion of Pluto by formalising the case study and applying the Reidealisation repair plan.

An outline of the axiomatisation of the case study is presented in Figure 8.13, where:

- *Planet*, *DPlanet* and *Unknown* are types of planets, dwarf planets and unknown objects;
- $(ClearNeighbour{::}\alpha \Rightarrow Bool)(o)$ returns true if the object o has cleared the neighbourhood around its orbit, i.e. satisfying c);
- $(Orbit :: \alpha \Rightarrow Mom \Rightarrow \{\langle\mathbb{R},\mathbb{R},\mathbb{R}\rangle\})(o)$ returns the orbit of the object o which is represented as a function taking a time moment and returning a three-dimensional point in space
- $(KuiperBelt{:}\{Unknown\}$ represents the Kuiper Belt as a set containing unknown celestial objects;
- *CoordA* and *CoordB* are two example coordinates;
- *PlutinoA* and *PlutinoB* are two example bodies within the Kuiper Belt; and,
- $(FromChart :: Obj \Rightarrow Mom \Rightarrow \langle Mom,\mathbb{R},\mathbb{R},\mathbb{R}\rangle)(z,t)$ returns a four-dimensional point occupied by z at t in space-time according to a sky chart.

In the theoretical ontology, we establish the type of *Pluto* as *Planet* (8.141). We have simplified the modelling of the necessary requirements for classification by considering only c), i.e. that an object has a clear neighbourhood if it has type *Planet* and does not have a clear neighbourhood if it has type *DPlanet* (8.142). A clear neighbourhood for an object means that no other celestial objects share any part of its orbit at any time (8.143). In the observation ontology, we assert that several plutinos are within the Kuiper Belt (8.145) and plutinos *PlutinoA* and *PlutinoB* are located at *CoordA* and *CoordB* at T_1, respectively (8.150). We also assert that Pluto locates at *CoordA* at T_2.

The function *FromChart* is not shared between *Pred* and *Obs*, so we are dealing with heterogeneous ontologies in this case study. We can identity an ontology in $\mathcal{F}(Pred)$ such that the merge between a sensory ontology, the bridging ontology, and this ontology does not induce inconsistency. Suppose this ontology is $Pred_f$:

$$
\begin{aligned}
Ax(Pred_f) \quad ::= \quad & \{ \\
& \forall x.ClearNeighbour(x) \longleftrightarrow \qquad (8.151) \\
& \forall x \neq y, t_1, t_2.Orbit(x,t_1) \neq Orbit(y,t_2) \\
& \}
\end{aligned}
$$

To reason with $Pred_f$, the following bridging axiom is required:

$$\forall x,t,v.Pred_f \vdash Orbit(x,t) = v \longleftrightarrow \tag{8.152}$$

$$Obs \vdash \langle \#2(FromChart(x,t)), \#3(FromChart(x,t)), \tag{8.153}$$

$$\#4(FromChart(x,t))\rangle = v \tag{8.154}$$

where $\#i$ returns the i^{th} element of a tuple.

From the formalisation of the case study, we can show the following:

$$Pred \vdash ClearNeighbour(Pluto) = True \tag{8.155}$$

$$Obs \oplus O_b \oplus Pred_f \vdash ClearNeighbour(Pluto) = False \tag{8.156}$$

$$\nu(Pred) \vdash ClearNeighbour(\nu(Pluto)) = False \tag{8.157}$$

$$\nu(Obs) \oplus \nu(O_b) \oplus \nu(Pred_f) \vdash ClearNeighbour(\nu(Pluto)) = False \tag{8.158}$$

where $\nu(Pred)$ is the repaired *Pred*; *Pluto* is of type *Planet* and $\nu(Pluto)$ is of type *DPlanet*; (8.155) is inferred from (8.142), given that $Pluto{:}Planet$; (8.156) can be proved because *Pluto*'s neighbourhood is not cleared according to observations; and (8.157) and (8.158) are provable using (8.142), given that $\nu(Pluto){:}DPlanet$.

We can instantiate the trigger formulae of the Reidealisation repair plan with the following substitution:

$$\{Pred/O_1, Obs \oplus O_b \oplus Pred_f/O_2, Pluto/stuff, ClearNeighbour/f, True/v_1, False/v_2, Planet/\tau_1, DPlanet/\tau_2\}$$

Following (4.34 - 4.37, p.4.34) of the repair plan, the corresponding repair is then:

$$\begin{aligned}
Sig(\nu(Pred)) &::= \{\nu(Pluto){:}DPlanet\} \cup Sig(Pred) \setminus \{Pluto\} \\
Sig(\nu(Obs)) &::= \{\nu(Pluto){:}DPlanet\} \cup Sig(Obs) \setminus \{Pluto\} \\
Ax(\nu(Pred)) &::= \{\phi\{\nu(Pluto)/Pluto\} \mid \phi \in Ax(Pred)\} \\
Ax(\nu(Obs)) &::= \{\phi\{\nu(Pluto)/Pluto\} \mid \phi \in Ax(Pred)\}
\end{aligned}$$

which means Pluto is now reclassified to be a dwarf planet, as required.

8.2.4.2 Test Case Study II: The Discovery of Denisovans

Fossils, including a finger bone, belonging to an unknown type of archaic human were found in the Denosiova cave in 2010. The mitochondrial DNA (mDNA) of the bone

$$Sig(Pred) \supseteq \{fingerbone{:}Neanderthals\} \tag{8.159}$$

$$Ax(Pred) \supseteq \{\forall x.\ KnownSpecies(x) = \begin{cases} True & \text{if } x{:}Neanderthals \text{ or} \\ & x{:}Homosapiens \\ False & \text{if } x{:}Unknown \end{cases}$$

$$\forall x,y.\ x \neq y \longrightarrow (mtDNA(x) \neq mtDNA(y) \longrightarrow \neg KnownSpecies(x)) \tag{8.160}$$

$$Ax(Obs) \supseteq \{\forall x.\ x \neq fingerbone \longrightarrow mtDNA(fingerbone) \neq mtDNA(x)\} \tag{8.161}$$

where *Pred* specifies the conditions in order for a specifies to be already known and *Obs* asserts that the mtDNA of the finger bone is unique.

Figure 8.14: Summary of the axiomatisation of the discovery of Denisovans case study.

fragments was shown to be distinct from those of Neanderthals and modern humans. The discovery team called this unknown type *Denisovans*, named after the cave in which the bones were found. We can emulate this episode of ontology evolution in biology by applying the Reidealisation repair plan to it.

A summary of the axiomatisation of the case study is shown in Figure 8.14, where:

- *fingerbone* represents the finger bone found in the Denisova Cave
- *Neanderthals* is the type of Neanderthals, *Homosapiens* is the type of Homosapiens, and *Unknown* is the type of unknown species
- $(KnownSpecies :: \alpha \Rightarrow Bool)(x)$ returns true if x comes from a Neanderthal and false if the source of x is unknown
- $(mtDNA :: \alpha \Rightarrow DNASeq)(x)$ returns the DNA sequence of some object x.

In the theoretical ontology, we define the type of *fingerbone* to be *Neanderthals* because Denisovans were initially believed to belong to Neanderthals or Homosapiens (8.159). We consider that an object comes from a known species if the source is a Neanderthal; if the source is unknown, then the object does not come from a known species (8.160). We also assert that if the mtDNA of an object is unique, then it does not come from a known species (8.160). In the observation ontology, we assert that the mtDNA of the finger bone is unique (8.161).

The formalisation of this case study is homogeneous, so there is no need for bridging axioms. We can show the following:

$$Pred \vdash KnownSpecies(mtDNA(fingerbone)) \tag{8.162}$$

$$Obs \oplus Pred_f \vdash \neg KnownSpecies(mtDNA(fingerbone)) \tag{8.163}$$

$$\nu(Pred) \vdash \neg KnownSpecies(mtDNA(\nu(fingerbone))) \tag{8.164}$$

$$\nu(Obs) \oplus \nu(Pred_f) \vdash \neg KnownSpecies(mtDNA(\nu(fingerbone))) \tag{8.165}$$

where $Pred_f$ is an ontology in the factorised network constructed from $Pred$, which contains only (8.160).

The trigger formulae of the Reidealisation repair plan can be instantiated as follows:

$$\{Pred/O_1, Obs \oplus Pred_f/O_2, fingerbone/stuff, \lambda x.KnownSpecies(mtDNA(x))/f,$$
$$True/v_1, False/v_2, Neanderthals/\tau_1, Unknown/\tau_2\}$$

Following the rule to effect the repair, the corresponding repair is therefore:

$$\begin{aligned}
Sig(\nu(Pred)) &::= \{\nu(fingerbone){:}Unknown\} \cup Sig(Pred) \setminus \{fingerbone\} \\
Sig(\nu(Obs)) &::= \{\nu(fingerbone){:}Unknown\} \cup Sig(Obs) \setminus \{fingerbone\} \\
Ax(\nu(Pred)) &::= Ax(Pred) \\
Ax(\nu(Obs)) &\supseteq \{\forall x.\, x \neq \nu(fingerbone) \longrightarrow mtDNA(\nu(fingerbone)) \neq mtDNA(x)\}
\end{aligned}$$

which means the finger bone found is now reclassified to have come from an unknown species rather than Neanderthals, as required.

8.2.4.3 Other Test Case Studies

In addition to the case studies presented in (§8.2.4.1) and (§8.2.4.2), we have also evaluated Reidealisation over a range of other case studies. These include:

- *Primordial Giant*: In 2007, a supernova was observed to last for an unusually long period – at least 555 days. With a supernova model, its mass was calculated to be 300 solar masses, which was almost two times the absolute upper limit on a modern star. However, in early cosmic times – immediately after the big bang – stars then lived briefly and died violently. This resembles a Reidealisation type of repair, because the idealisation of the supernova as a modern star caused a contradiction about its mass, which disappears if the star is reidealised as a primordial star.

- *Two Faces of Water*: Water behaves differently from most other substances in various ways, including that it is less dense at its freezing point ($0°C$) than it is above. Some physicists postulate that the disparity comes from water taking two forms, which behave differently and the observed behaviour is the interaction of these two. One is a crystalline form, which ice takes, and the other disordered. It turns out that water molecules are more densely packed when they are in a disordered structure. In essence, the repair is to not idealise water as a substance that can be packed in only one way but in two fundamentally different ways.

- *Large-Scale Cosmic Magnetic Fields*: The polarisation of light is known to be twisted by large-scale magnetic fields, but the source of their seed field is unexplained by conventional mechanisms. With the theory of dark magnetism, which postulates new and unobserved magnetic fields. So, the repair is to not idealise such large-scale cosmic magnetic fields as products of conventional physics, but of the theory of dark magnetism instead.

8.2.4.4 Summary

Reidealisation address a kind of fault similar to WMS, but it requires a stronger ontological setup which requires the ontologies to consider more than one idealisation of some property. A resulting benefit is that the repair entails reassigning the type of the concept responsible for creating the fault, which provides more meaningful repair than creating an invisible component.

8.2.5 Spectrum

Unary predicates can be thought of as set memberships in a way that the name of the predicate corresponds to the name of a set and the predicate is true if and only if the argument is a member of the set. This is often sufficient for working in simple and controlled problem domains, but is a relatively unnatural representation for domains involving complex relationships between concepts, such as Physics. The Spectrum ontology repair plan is designed to create a new structure that provides additional logical power from unary predicates.

	Solid	Liquid	Gas	Plasma
Solid	–	Melting	Sublimation	–
Liquid	Freezing	–	Evaporation	–
Gas	Deposition	Condensation	–	Ionisation
Plasma	–	–	Deionisation	–

Table 8.1: Types of phase transition.

8.2.5.1 Test Case Study I: Nomenclature for Phase Transitions

A phase transition is a change of a substance from one state to another. In §4.3.1, we have considered the heat required by a substance in order to undergo phase transition, whereas this case study deals with the actual types of phase transition. More precisely, latent heat case study describes the essential episode that causes phase transitions, the discovery of which can be repaired by WMS, as demonstrated. We explain below that the effect produced by the latent heat case study can be repaired by the Spectrum repair plan.

Typically, a phase transition is used to describe transitions between solid, liquid, gaseous and plasma states. A particular phase transition happens at a specific temperature. However, a phase transition is a result of a change of some external condition and what is modelled here is the effect of a change of temperature. The types of transition between solid, liquid, gas and plasma are shown in Table 8.1. Suppose each state is represented as an unary predicate, taking a triple consisting of the initial temperature, the final temperature and the substance under consideration as argument:

$$(Melting :: \langle \mathbb{R}, \mathbb{R}, Obj \rangle \Rightarrow Bool)(\langle t_1, t_2, s \rangle) \tag{8.166}$$

$$(Freezing :: \langle \mathbb{R}, \mathbb{R}, Obj \rangle \Rightarrow Bool)(\langle t_1, t_2, s \rangle) \tag{8.167}$$

$$(Evaporation :: \langle \mathbb{R}, \mathbb{R}, Obj \rangle \Rightarrow Bool)(\langle t_1, t_2, s \rangle) \tag{8.168}$$

$$(Condensation :: \langle \mathbb{R}, \mathbb{R}, Obj \rangle \Rightarrow Bool)(\langle t_1, t_2, s \rangle) \tag{8.169}$$

$$(Sublimation :: \langle \mathbb{R}, \mathbb{R}, Obj \rangle \Rightarrow Bool)(\langle t_1, t_2, s \rangle) \tag{8.170}$$

$$(Deposition :: \langle \mathbb{R}, \mathbb{R}, Obj \rangle \Rightarrow Bool)(\langle t_1, t_2, s \rangle) \tag{8.171}$$

$$(Ionisation :: \langle \mathbb{R}, \mathbb{R}, Obj \rangle \Rightarrow Bool)(\langle t_1, t_2, s \rangle) \tag{8.172}$$

$$(Deionisation :: \langle \mathbb{R}, \mathbb{R}, Obj \rangle \Rightarrow Bool)(\langle t_1, t_2, s \rangle) \tag{8.173}$$

where (8.166) is true when substance s undergoes melting when its initial and final temperatures are t_1 and t_2, respectively; (8.167) is true when substance s undergoes freezing when its initial and final temperatures are t_1 and t_2, respectively; and the rest are self-explanatory. Suppose an ontology *Ont* specifies the type of phase transition for water and nitrogen[14] over some temperature ranges[15], the case study can be axiomatised as Figure 8.15. The axiomatisation specifies the types of phase transition for various observations, e.g., when the initial and final temperatures of some water are -1C and 1C, respectively (8.175). Note that all predicates use mnemonic names, so that the set of related predicates could be constructed by measuring the semantic distances between each predicate in the language, which could, for instance, be computed using WordNet (Miller, 1995). It is not unreasonable to assume that *Melting*, *Freezing*, *Evaporation*, and so forth, could be regarded to be semantically related by some algorithm. If the language contains a predicate named, say, *Rough*, then it is also not unreasonable to assume that it would be regarded to be semantically unrelated, because *rough* is typically associated with the texture of material rather than the physical transition in phase.

Even with the set $RelatedPredicates$ constructed, we only know that its members are semantically related to a certain extent, but their relationships have no logical meaning. To increase the logical power, the Spectrum repair plan (Figure 4.7, p.68) can be applied to produce a new representation. The Spectrum repair plan can be triggered by the following substitution:

$$\{Ont/O, Observations/Q, RelatedPredicates/P\} \tag{8.187}$$

as $Observations$ and $RelatedPredicates$ are already maximal subsets that do satisfy the trigger formula.

To effect the repair in the signature of *Ont*, we declare the following new symbols:

$$PhaseTransition{:}\mathbb{R} \times \mathbb{R} \times Obj \Rightarrow TransitionType \tag{8.188}$$

$$Melting{:}TransitionType \tag{8.189}$$

$$Freezing{:}TransitionType \tag{8.190}$$

$$Evaporation{:}TransitionType \tag{8.191}$$

$$Condensation{:}TransitionType \tag{8.192}$$

$$\ldots$$

$$Deionisation{:}TransitionType \tag{8.193}$$

[14]The calculation of the temperatures under which ionisation or deionisation occurs is achieved by using the equation for the thermal energy of the ideal gas and nitrogen's first ionisation energy of 1402.3 kJ/mol.

[15]Standard atmospheric pressure is assumed in order to simplify the underlying Physics.

$$Sig(Ont) ::= \{(8.166),(8.167),(8.168),(8.169),(8.170),(8.171) \quad (8.174)$$
$$(8.172),(8.173), H_2O{:}Obj, N{:}Obj\}$$
$$Ax(Ont) ::= \{Melting(\langle -1, 1, H_2O\rangle), \quad (8.175)$$
$$Freezing(\langle 1, -1, H_2O\rangle), \quad (8.176)$$
$$Evaporation(\langle 99, 101, H_2O\rangle), \quad (8.177)$$
$$Condensation(\langle 101, 99, H_2O\rangle), \quad (8.178)$$
$$Melting(\langle -220, -200, N\rangle), \quad (8.179)$$
$$Freezing(\langle -200, -220, N\rangle), \quad (8.180)$$
$$Evaporation(\langle -200, -190, N\rangle), \quad (8.181)$$
$$Condensation(\langle -190, -200, N\rangle), \quad (8.182)$$
$$Ionisation(\langle 90000, 100000, N\rangle), \quad (8.183)$$
$$Deionisation(\langle 100000, 90000, N\rangle), \quad (8.184)$$
$$Observations ::= \{\langle -1, 1, H_2O\rangle, \langle 1, -1, H_2O\rangle, \langle 99, 101, H_2O\rangle, \quad (8.185)$$
$$\langle 101, 99, H_2O\rangle \ldots\}$$
$$RelatedPredicates ::= \{Melting, Freezing, Evaporation, \quad (8.186)$$
$$Condensation, Deposition, Ionisation, \ldots\}\}$$

where the ontology *Observations* is a set containing all targets of observation expressed as triples and *RelatedPredicates* is a set containing all predicates that are deemed related. All predicates here use mnemonic names, so the set *RelatedPredicates* can be constructed by computing the semantic distances between each predicate in the language using WordNet-based or more advanced machine learning-based approaches.

Figure 8.15: Axiomatisation of the nomenclature of phase transitions.

and remove the corresponding old symbols. Thus, the new signature of Ont, $Sig(\nu(Ont))$, is:

$$Sig(\nu(Ont)) ::= \{Melting{:}TransitionType, \tag{8.194}$$
$$Freezing{:}TransitionType, Evaporation{:}TransitionType, \tag{8.195}$$
$$Condensation{:}TransitionType, Sublimation{:}TransitionType, \tag{8.196}$$
$$Deposition{:}TransitionType, Ionisation{:}TransitionType, \tag{8.197}$$
$$Deionisation{:}TransitionType, H_2O{:}Obj, N{:}Obj, \tag{8.198}$$
$$PhaseTransition{:}\langle \mathbb{R} \Rightarrow \mathbb{R} \Rightarrow Obj\rangle \Rightarrow TransitionType\} \tag{8.199}$$

The axioms are repaired as follows:

$$Ax(\nu(Ont)) ::= \{PhaseTransition(\langle -1, 1, H_2O\rangle) = Melting, \tag{8.200}$$
$$PhaseTransition(\langle 1, -1, H_2O\rangle) = Freezing, \tag{8.201}$$
$$PhaseTransition(\langle 99, 101, H_2O\rangle) = Evaporation, \tag{8.202}$$
$$PhaseTransition(\langle 101, 99, H_2O\rangle) = Condensation, \tag{8.203}$$
$$PhaseTransition(\langle -220, -200, N\rangle) = Melting, \tag{8.204}$$
$$PhaseTransition(\langle -200, -220, N\rangle) = Freezing, \tag{8.205}$$
$$PhaseTransition(\langle -200, -190, N\rangle) = Evaporation, \tag{8.206}$$
$$PhaseTransition(\langle -190, -200, N\rangle) = Condensation, \tag{8.207}$$
$$PhaseTransition(\langle 90000, 100000, N\rangle) = Ionisation, \tag{8.208}$$
$$PhaseTransition(\langle 100000, 90000, N\rangle) = Deionisation \tag{8.209}$$
$$\ldots\}$$

With these new axioms, there is only one type of phase transition associated with a substance and a temperature range by the injection $PhaseTransition$, provided that we exclude from our model situations where the substance under consideration undergoes multiple phase changes. So a structure for types of phase transitions is constructed, as required.

8.2.5.2 Test Case Study II: Construction of Cosmic Distance Ladder

Estimating distances between Earth and celestial objects is notoriously complex because of the vast scale of the universe. The closest star to Earth, Alpha Centauri A, is 4.3 light years away, whilst the most distant galaxies yet observed are more than 13 billion light years away. To measure these distances, astronomers adopt a variety of techniques:

- Direct distance measurements are only possible for objects within a distance of about 1,000 light-years even with precision, space-based telescopes. For these relatively nearby objects, astronomers use a phenomenon called *parallax*, which is when an object appears to move relative to a more distant background due to the motion of the observer.

- To determine the distances to more distant objects, *Cepheid variables* are useful for measuring objects out to a maximum distance of 60 million light-years. Cepheid variables are a class of stars that pulse in and out, and the rate of their pulses are governed by their brightnesses. The distance can be calculated by using parallax to determine the distances to nearby Cepheids and comparing the true brightness to the apparent brightness of the object.

- Stars of the class *Type Ia supernovae* can be used to measure objects out to a few billion light-years away. These are exploding stars that brighten and fade in a way that their absolute magnitudes are all the same, which reveals their distances.

- For objects as far as 13 billion light-years away, *redshift* measures the speed galaxies are moving away from Earth. Because the universe is constantly expanding, distant galaxies move away faster creating the doppler effect. Note that the case study in §8.2.1.2 is an example of determining distance using redshift.

These four methods form part of the *cosmic distance ladder*, as the entire structure presenting the methods is analogous to a ladder, where each rung corresponds to a range of distances.

Suppose each technique for measuring astronomical distances is represented as an unary predicate, taking a pair consisting of the observation event of measurement and the celestial object in consideration as argument:

$$(\mathit{Parallax}{::}\langle \mathit{Event}, \mathit{Obj}\rangle \Rightarrow \mathit{Bool})(\langle e, o\rangle) \tag{8.210}$$

$$(\mathit{Cepheids}{::}\langle \mathit{Event}, \mathit{Obj}\rangle \Rightarrow \mathit{Bool})(\langle e, o\rangle) \tag{8.211}$$

$$(\mathit{Supernovae}{::}\langle \mathit{Event}, \mathit{Obj}\rangle \Rightarrow \mathit{Bool})(\langle e, o\rangle) \tag{8.212}$$

$$(\mathit{Redshift}{::}\langle \mathit{Event}, \mathit{Obj}\rangle \Rightarrow \mathit{Bool})(\langle e, o\rangle) \tag{8.213}$$

where (8.210) is true when the distance to object o can be measured accurately by the Parallax method at the observation event e; (8.211) is true when the distance to object o can be measured accurately by the Cepheids method at the observation event e; (8.212) is true when the distance to object o can be measured accurately by the Type Ia Supernovae method at the observation event e; and, (8.213) is true when the distance to object o can be measured accurately by the Redshift method at the observation event e. Suppose an ontology *Ont* specifies the accurate measuring used for various celestial objects, including:

- Sirius, which is 8.6 light-years away
- Andromeda Galaxy, which 2.5 million light-years away
- Supernova 1994D (SN1994D), which is 108 million light-years away
- IOK-1, which is 13 billion light-years away.

The case study can be axiomatised as Figure 8.16, which specifies the types of measurement that is accurate for each object at the corresponding observation event, e.g., the distance to Sirius can be accurately measured by the Parallax method (8.215). Note that all predicates use mnemonic names, so the set of related predicates $RelatedPredicates$ can be constructed by measuring the semantic distances between each predicate in the language, similar to the previous case study.

Even with the set $RelatedPredicates$ constructed, the relationships between each member have no logical meaning. The Spectrum repair plan can be triggered by the following substitution:

$$\{Ont/O, Observations/Q, RelatedPredicates/P\} \tag{8.221}$$

as *Observations* and $RelatedPredicates$ are already maximal subsets that do satisfy the trigger formula.

To effect the repair in the signature of *Ont*, we declare the following new symbols:

$$DistanceMethod{:}\langle Event \Rightarrow Obj\rangle \Rightarrow Method \tag{8.222}$$

$$Parallax{:}Method \tag{8.223}$$

$$Cepheids{:}Method \tag{8.224}$$

$$Supernovae{:}Method \tag{8.225}$$

$$Redshift{:}Method \tag{8.226}$$

$$Sig(Ont) ::= \{(8.210),(8.211),(8.212),(8.213),Sirius{:}Obj, \quad (8.214)$$

$$Andromeda{:}Obj, SN1994D{:}Obj, IOK-1{:}Obj, ObsA{:}Event,$$

$$ObsB{:}Event, ObsC{:}Event, ObsD{:}Event\}$$

$$Ax(Ont) ::= \{Parallax(\langle ObsA, Sirius\rangle), \quad (8.215)$$

$$Cepheids(\langle ObsB, Andromeda\rangle), \quad (8.216)$$

$$Supernovae(\langle ObsC, SN1994D\rangle), \quad (8.217)$$

$$Redshift(\langle ObsD, IOK-1\rangle), \quad (8.218)$$

$$Observations ::= \{\langle ObsA, Sirius\rangle, \langle ObsB, Andromeda\rangle, \langle ObsC, SN1994D\rangle, \quad (8.219)$$

$$\langle ObsD, IOK-1\rangle \ldots\}$$

$$RelatedPredicates ::= \{Parallax, Cepheids, Supernovae, Redshift\}\} \quad (8.220)$$

where the ontology $Observations$ is a set containing all targets of observation expressed as pairs and $RelatedPredicates$ is a set containing all predicates that are deemed related. All predicates here use mnemonic names, so the set $RelatedPredicates$ can be constructed by computing the semantic distances between each predicate in the language using WordNet-based or more advanced machine learning-based approaches.

Figure 8.16: Axiomatisation of the construction of cosmic distance ladder.

and remove the corresponding old symbols. Thus, the new signature of Ont, $Sig(\nu(Ont))$, is:

$$Sig(\nu(Ont)) ::= \{Parallax{:}TransitionType, \quad (8.227)$$

$$Cepheids{:}Method, Supernovae{:}Method, \quad (8.228)$$

$$Redshift{:}Method, Sirius{:}Obj, Andromeda{:}Obj, SN1994D{:}Obj \quad (8.229)$$

$$IOK-1{:}Obj, DistanceMethod{:}\langle Event \Rightarrow Obj\rangle \Rightarrow Method\} \quad (8.230)$$

The axioms are repaired as follows:

$$Ax(\nu(Ont)) ::= \{DistanceMethod(\langle ObsA, Sirius\rangle) = Parallax, \quad (8.231)$$

$$DistanceMethod(\langle ObsB, Andromeda\rangle) = Cepheids, \quad (8.232)$$

$$DistanceMethod(\langle ObsC, SN1994D\rangle) = Supernovae, \quad (8.233)$$

$$DistanceMethod(\langle ObsD, IOK-1\rangle) = Redshift, \quad (8.234)$$

$$\ldots\}$$

This is precisely the invention of the cosmic distance ladder from the unary predicate representation. However, the target type of $DistanceMethod$ (8.222), i.e. τ' in (Figure 4.7, p.68), is the unordered nominal type $Method$, so it does not yet reveal the desirable ladder structure. A further step is needed to define a partial order based on the distances at each rung.

8.2.5.3 Summary

Spectrum could be viewed as a variation of *Unite*, as Unite merges two different concepts into one, whereas Spectrum identifies similar concepts and groups them together to form a new structure. Instead of measuring the defining property of some concept, Spectrum looks for functional properties in its original representation.

8.3 Alternative Theories

In this section, we look at some of the other instantiations retained by the heuristic filters. In particular, we are interested in understanding the plausibility of repairing using these instantiations. If some of the unfiltered instantiations are physically plausible, then GALILEO would demonstrate creative behaviour, as these instantiations were unanticipated by its creators. To avoid excessive verbosity, we describe the alternative theories only for the WMS and Inconstancy case studies that are reported earlier.

8.3.1 "Where's My Stuff"

8.3.1.1 The Bouncing-Ball Paradox

Described in §8.2.1.1 is the bouncing-ball paradox, which is demonstrated to be repairable by WMS. Recall that in this case study, the following constants occur:

- $TE{:}Obj \Rightarrow Mom \Rightarrow \mathbb{R}$ is the total energy of an object.
- $Ball{:}Obj$ is the bouncing ball.
- $Drop{:}Event$ is the event of dropping the ball.
- $End{:}Event \Rightarrow Mom$ is the final moment of an event.

Because we work with the contradiction outlined in (8.23, 8.24), $f(stuff)$ is restricted to $TE(Ball, End(Drop))$ in each instantiation, i.e. the total energy of the ball at the moment that it hits the ground. The reason is that it is this exact formula that creates a conflict, at least in the case study – the value of $TE(Ball, End(Drop))$ is different in the conflicting ontologies. After applying the heuristic filters, 10 instantiations are returned, as listed below:

$$?f ::= \lambda a.TE(a,(End(Drop)) \qquad ?stuff ::= Ball \tag{8.235}$$

$$?f ::= \lambda a.TE(Ball,(End(a))) \qquad ?stuff ::= Drop \tag{8.236}$$

$$?f ::= TE(Ball) \qquad ?stuff ::= End(Drop) \tag{8.237}$$

$$?f ::= \lambda a.TE(Ball,(aDrop)) \qquad ?stuff ::= End \tag{8.238}$$

$$?f ::= \lambda a.a \qquad ?stuff ::= TE(Ball,(End(Drop))) \tag{8.239}$$

$$?f ::= \lambda a.a(End(Drop)) \qquad ?stuff ::= TE(Ball) \tag{8.240}$$

$$?f ::= \lambda a.a(Ball) \qquad ?stuff ::= \lambda a.TE(a,End(Drop)) \tag{8.241}$$

$$?f ::= \lambda a.a(Drop) \qquad ?stuff ::= \lambda a.TE(Ball,End(a)) \tag{8.242}$$

$$?f ::= \lambda a.a(Ball,End(Drop)) \qquad ?stuff ::= TE \tag{8.243}$$

$$?f ::= \lambda a.a(Ball,Drop) \qquad ?stuff ::= \lambda a,b.TE(a,End(b)) \tag{8.244}$$

The preferred instantiation of *stuff* is *TE*, because the case study requires the concept of total energy to be split into parts, in which case the instantiation of f is $\lambda a.\ a(Ball, End(Drop))$ (8.243). We argue that all of the other 9 instantiations are physically meaningful, but with different degrees of plausibilities.

8.3.1.1.1 Phlogiston? The instantiation (8.235) suggests that the ball contains an invisible part which temporarily stores the missing energy. The basics of this hypothesis are similar to those of the *phlogiston* theory: an obsolete theory that postulated that all combustible materials contain some substance that is released in burning. Our phlogiston would act like a store for kinetic energy, which is given off to power the rebound. Although modern science has refuted the phlogiston theory, the phlogiston theory was a respectable theory, which just happened to be wrong. As we have mentioned throughout the thesis, it is not the job of our ORPs to restrict themselves to only *historically correct* repairs, but to generate *plausible* ones.

8.3.1.1.2 Many-worlds interpretation? The variable *stuff* is instantiated to *End* in (8.238), which is a function that returns the final moment of an event. An approach to quantum mechanics is the many-worlds interpretation, which asserts that, in addition to the world we are aware of directly, there are other similar worlds existing in parallel. The many-worlds interpretation is used to explain the 'Schrodinger's cat' paradox, which presents a scenario where a cat might be both alive and dead, and postulates that every event is a branch point and the alive and dead cats are in different branches of the universe. Our Schrodinger's cat would be the ball and, by the many-worlds interpretation, the total energy of the ball could be greater than zero in one world and zero in another. Let us suppose that End_A returns the final moment of an event in the branch A where the ball is stationary upon impact and that End_B returns the final moment of an event another branch, B. The repair is to split *End* into End_A and End_B. Now at the moment returned by End_A, the ball is indeed stationary, so no contradiction. This story implies that if the total *stuff* represents the classical view of reality, then it can be indirectly expressed as the summation of some representation of time in two parallel worlds.

A similar story can also be devised for instantiations (8.236) and (8.237).

8.3.1.1.3 Exceptional object? The instantiation (8.240) has *stuff* instantiated to $TE(Ball)$, which means that *TE* and *Ball* do not warrant repair in general, but only when *TE* is applied to *Ball*. This is somewhat similar to Lakatos's exception-barring methods, e.g., *piecemeal exclusion* (Lakatos, 1976), but we determine an extension of the set of counter-examples without explicitly constructing a set containing counter-examples. In our case, the original definition of *TE* is still applicable to every object within the domain that is not *Ball*, and for *Ball* a new definition is used. One plausible reason for this is that the structure of *Ball* is exceptional, e.g., the original definition of *TE* is already a good approximation – similar to how classical mechanics are still widely adopted today even when quantum mechanics have been accepted by modern science.

A similar story can be concocted for (8.241), (8.242) and (8.244):

- In (8.241), instead of excluding *Ball* from the set of supporting examples, the final moment of an event is excluded. A plausible reason is that all objects give off "phlogiston" at the final moment of the dropping event.
- In (8.242), the particular *Ball* has an exceptional behaviour at the final moment

of all events. One explanation is that the "phlogiston" gives off only at the end of an event.

- In (8.244), all objects give off phlogiston at the end of all events.

8.3.1.1.4 Experimental error? Instantiation (8.239) is relatively specific, as it suggests that the conflict does not lie in the definition or conceptualisation of *TE*, *Ball*, etc. individually, but in the application of *TE* on *Ball* and $End(Drop)$. We can interpret this to be an experimental error with this particular measurement rather than a more general problem with the laws or conceptualisation.

8.3.1.2 Dark Matter

Described in §8.2.1.2 is the dark matter case study, which is also demonstrated to be repairable by WMS. Recall that in this case study, the following constants occur:

- $GraphA{:}Glxy \Rightarrow Mom \Rightarrow Matter \Rightarrow \langle \mathbb{R}, \mathbb{R} \rangle$ is the rotation curve of a galaxy.
- $glxy101{:}Glxy$ is the galaxy under consideration.
- $obs1{:}Event$ is the observation event.

Because we work with the paradox formulated in (8.51, 8.52), $f(stuff)$ is restricted to $GraphA(glxy101, obs1)$ in each instantiation, i.e. the rotation curve of the galaxy $glxy101$ based on the observation event $obs1$. The reason is that it is this exact formula that creates a conflict, at least in the case study – the value of $GraphA(glxy101, obs1)$ is different in the conflicting ontologies. After applying the heuristic filters, 5 instantiations are returned, as listed below:

$$?f ::= \lambda a.GraphA(a, obs1) \qquad ?stuff ::= glxy101 \tag{8.245}$$
$$?f ::= GraphA(glxy101) \qquad ?stuff ::= obs1 \tag{8.246}$$
$$?f ::= \lambda a.a \qquad ?stuff ::= GraphA(glxy101, obs1) \tag{8.247}$$
$$?f ::= \lambda a.a(glxy101) \qquad ?stuff ::= \lambda a.GraphA(a, obs1) \tag{8.248}$$
$$?f ::= \lambda a.a(obs1) \qquad ?stuff ::= GraphA(glxy101) \tag{8.249}$$

The preferred instantiation of *stuff* is $glxy101$, because the case study requires the representation of a galaxy to be split into parts, in which case the instantiation of f is $\lambda a.\ GraphA(a, obs1)$ (8.245). We argue that all of the other 4 instantiations are physically meaningful, but with different degrees of plausibilities.

8.3.1.2.1 Many-worlds interpretation? The variable *stuff* is instantiated to $obs1$ in (8.246), which is an observation event. Similar to the bouncing-ball, splitting $obs1$ into parts could be interpreted as adopting the many-worlds interpretation of quantum mechanics. Let us suppose that $obs1_A$ returns the observation event occurred in the branch A where the orbital speeds of distant stars are faster than those observed in the classical interpretation of real world. Also, let us suppose that $obs1_B$ returns the observation event in another branch, B. The repair is to split $obs1$ into $obs1_A$ and $obs1_B$. Now according to the observation event $obs1_A$, the orbital speeds are indeed greater, so there is no contradiction because $obs1_A$ deals with a different branch of the universe.

8.3.1.2.2 Exceptional object? The instantiation (8.249) has *stuff* instantiated to $GraphA(glxy101)$, which means that $GraphA$ and $glxy101$ do not warrant repair in general, but only when $GraphA$ is applied to $glxy101l$. Like in the bouncing-ball paradox, this is somewhat similar to Lakatos's method of piecemeal exclusion as well. In our case, the original definition of $GraphA$ is still applicable to every object within the domain that is not $glxy101$, and for $glxy101$ a new definition is used. One plausible reason for this is that the structure of $glxy101$ is exceptional, e.g., only $glxy101$ experiences the effects of dark matter.

A similar story can be concocted for (8.248).

8.3.1.2.3 Experimental error? Instantiation (8.247) is relatively specific. It suggests that the conflict does not lie in the definition or conceptualisation of $GraphA$, $glxy101$, etc. individually, but in the application of $GraphA$ on $glxy101$ and $obs1$. We can interpret this to suggest an experimental error in this particular measurement rather than a more general problem with the definitions or conceptualisations, due to the high specificity.

8.3.2 Inconstancy

8.3.2.1 The Travel Time of Light

Described in §8.2.2.1 is the travel-time of light case study, which is demonstrated to be repairable by the Inconstancy repair plan. In this case study, several constants occur, including:

- $Approx$:$\mathbb{R} \Rightarrow \mathbb{R}$ returns the approximated form of a value;
- $TravelTime$:$Obj \Rightarrow Point \Rightarrow Point \Rightarrow \mathbb{R}$ returns the time an object takes to travel to some point in space;
- $LightFrom$:$Obj \Rightarrow Obj$ returns the light object emitted from an object;
- $ReactionTime$:$Animal \Rightarrow \mathbb{R}$ returns the least number of seconds a typical animal responds to an event; and,
- $roemer$:$Person$ represents the observer conducting the observation.

Because we work with the paradox outlined in (8.72 - 8.73), applying the instantiation of $?f$ to that of $?stuff$ is guaranteed to give

$$Approx(TravelTime(LightFrom(Io), Io, Earth)) \tag{8.250}$$

in each match, i.e. the approximated form of the travel time of the light emitted by Io travelling to Earth. The reason is that it is this exact formula that creates a conflict – the values unexpectedly vary in the conflicting ontologies. After applying the heuristic filters, 43 matches containing unique instantiations of $?f$ and $?stuff$ remain. This is considerably more than the previous case study, due to the fact that there are substantially more terms occurring in (8.250) than in previous theorems.

We depict some matches below and all of the matches are shown in Figure A.1.

$$\begin{aligned} ?f &\mapsto Approx\ , \\ ?stuff &\mapsto TravelTime\ (LightFrom\ Io)\ Io\ Earth \end{aligned} \tag{8.251}$$

$$\begin{aligned} ?f &\mapsto \lambda a.\ Approx\ (a\ Io\ Earth)\ , \\ ?stuff &\mapsto \lambda a, b.\ TravelTime\ (LightFrom\ a)\ a\ b \end{aligned} \tag{8.252}$$

$$\begin{aligned} ?f &\mapsto \lambda\, a.\ Approx\ (TravelTime\ (LightFrom\ a)\ Io\ Earth)\ , \\ ?stuff &\mapsto Io \end{aligned} \tag{8.253}$$

$$\begin{aligned} ?f &\mapsto \lambda\, a.\ Approx\ (TravelTime\ (LightFrom\ Io)\ a\ Earth)\ , \\ ?stuff &\mapsto Io \end{aligned} \tag{8.254}$$

$$\begin{aligned} ?f &\mapsto \lambda\, a.\ Approx\ (TravelTime\ (LightFrom\ Io)\ Io\ a)\ , \\ ?stuff &\mapsto Earth \end{aligned} \tag{8.255}$$

$$\begin{aligned} ?f &\mapsto \lambda\, a.\ Approx\ (TravelTime\ a\ Io\ Earth)\ , \\ ?stuff &\mapsto LightFrom\ Io \end{aligned} \tag{8.256}$$

The preferred instantiation of *stuff* is $TravelTime(LightFrom(Io), Io, Earth)$ (8.251), because the case study requires the function returning the travel-time of the light emitted from Io travelled to Earth to be given a new dependency (8.76). We argue that all of the other 42 instantiations are physically meaningful, but with different degrees of plausibilities. In fact, instantiating *stuff* to $\lambda a, b.TravelTime(LightFrom(a), a, b)$ (8.252) will give a more general repair, because that will suggest the term to be given the variad is the travel time of light, disregarding the source and destination.

8.3.2.1.1 Wormhole? The matches (8.253 - 8.255) have *stuff* instantiated to either Io or Earth, respectively. This means Io or Earth will be given the variad as an additional argument, turning the terms representing Io and Earth into functions from constants. This is reminiscent of postulating about the existence of wormholes, which is virtually a shortcut through spacetime. If there is a wormhole connecting Io and Earth, then the light emitting off Io could travel through this tunnel. The property of the wormhole might change, which explains a variation in the value of the light's time of travel. Notice that both (8.253) and (8.254) have ?*stuff* instantiated to *Io*, but the instantiations of $?f$ are different. This is a particularly interesting result, because (8.253) suggests that the thing that unexpected changed is the light source whereas (8.254) suggests that it is the physical location of the source.

8.3.2.1.2 Changing properties of light? The match (8.256) instantiates *stuff* to the $LightFrom(Io)$, meaning that the light emitted from Io itself varies when the value of the variad changes. Representing the light beam itself as a function that takes the variad as the argument could mean that the physical state or property of the light changes, which consequently affects the time measured. This is similar to the postulate that light exhibits wave and particle properties. Certain behaviours of light are better explained by wave properties, e.g., double-slit interferences, whilst others by particle properties, e.g., refraction. This repair suggests that the unexpected variation in the measurements is solely caused by the physical change of light emitted from Io itself.

Most of the remaining instantiations can be interpreted by the many-worlds interpretation, conceptual problem, and experimental error explanations, similar to previous case studies.

8.3.2.2 Gas Laws

Described in §8.2.2.2 is the gas laws case study, which is demonstrated to be repairable by the Inconstancy repair plan. In this case study, several constants occur, including:

- $Boyle(g)$ returns a graph plotting time against the product of the pressure and volume of the gas g;
- $Hg(o,m)$ returns the mercury content with respect to an object o, according to the modifier m; if m is of value In, then it returns the mercury content *inside* o;
- $Flask(o,m)$ returns the flask with respect to object o, according to the modifier m; if m is of value On, then it returns the flask placed *on* o; and,
- $Desk$ is a desk object in the laboratory.

We work with the paradox outlined in (8.95 - 8.99), applying the instantiation of $?f$ to that of $?stuff$ is guaranteed to give

$$Boyle(Hg(Flask(Desk,On),In)) \tag{8.257}$$

in each match, i.e. the graph plotting time against Boyle's Law value of the gas that is placed inside a flask, which itself is placed on a desk. After applying the heuristic filters, 29 matches containing unique instantiations of $?f$ and $?stuff$ remain.

We highlight three of the matches below; all of the matches found are shown in Figure A.2.

$$?f \mapsto \lambda\, a.\, a\ (Hg\ (Flask\ Desk\ On)\ In), \tag{8.258}$$
$$?stuff \mapsto Boyle$$
$$?f \mapsto \lambda a.\, Boyle\ (Hg\ a\ In), \tag{8.259}$$
$$?stuff \mapsto Flask\ Desk\ On$$
$$?f \mapsto \lambda a.\, Boyle\ (Hg\ (a\ Desk)\ In), \tag{8.260}$$
$$?stuff \mapsto \lambda a.\, Flask\ a\ On$$

The preferred instantiation of *stuff* is *Boyle* (8.258), because the case study requires Boyle's Law to be given a new dependency. We argue that all of the other instantiations are physically meaningful, but with different degrees of plausibilities.

8.3.2.2.1 Irrelevant Measurements? The variable $?stuff$ is instantiated to

$$Flask(Desk, On) \tag{8.261}$$

in (8.259), suggesting that the particular flask placed on top of the desk has varying properties that affect the value of Boyle's Law. This might mean that the volume of space inside the flask changed during the experiment. One possibility is that the thickness of the flask itself increased/decreased, so it would be insufficient to base Boyle's value on the exterior dimensions of the flask. Alternatively, this particular flask cracked during the experiment when temperature was changed.

8.3.2.2.2 Unreliable Apparatus? The variable $?stuff$ is instantiated to

$$\lambda a.Flask(a, On) \tag{8.262}$$

in (8.260). It essentially represents a function that returns some flask that is placed on top of some given object. If the argument is a desk, then it returns the flask on top of the desk. Making this function depend on the temperature may mean that all flasks that are placed on top of anything have certain properties changed with temperature. For instance, the part of the flask making contact with another object when placed on top cracks at higher temperatures.

Similar to the previous case study, most of the remaining instantiations can be interpreted by the many-worlds interpretation, conceptual problem, and experimental error explanations.

8.3.3 Summary

We have outlined some interpretations of diagnoses other than the historically correct one returned by GALILEO. With the four filtering heuristics, we have successfully filtered out physically meaningless instantiations of $?stuff$. Each of the remaining instantiation corresponds to a plausible alternative theory.

8.4 Summary

GALILEO has been evaluated on a diverse collection of case studies from Physics, ranging from school-level Physics to advanced Astronomy. We aimed to verify the main hypothesis that the ORPs designed are indeed highly general and a few ORPs can account for a large number of instances of ontology evolution in Physics. In each case study, the detected ontological fault has successfully been resolved by the application of an appropriate ORP. These results indicate that GALILEO is already capable of performing well in the Physics domain, which is a considerably complex domain given the underlying higher-order nature.

We have verified that historically accurate solutions are within the search space in all case studies, which means that we have successfully emulated all episodes of ontology evolution in Physics that we considered. We have also explored beyond these solutions and investigated other plausible repairs. The four heuristics presented in §6.3 are sufficient to substantially eliminate physically meaningless diagnoses, allowing the selection of plausible repairs to become manageable. Each of the filtered diagnoses gives hints to new, interesting Physics, which enables GALILEO to assist the user better understand the underlying cause of the fault in question and even derive new Physics.

Chapter 9

Conclusions

9.1 Introduction

In this thesis, we have outlined the urgent need for improving the automation in ontology evolution in distributed environments. We have introduced and provided a description of GALILEO and presented a study of its techniques for ontology evolution. Our evaluation approach focused on measuring the applicability of the system and its techniques to a range of case studies from Physics. Further, throughout the thesis, we have suggested various areas of further work. We now discuss in more depth the ones of particular importance and significance. We outline possible directions for improving and extending the system and its techniques. Moreover, we summarise the work in this thesis, highlighting the central contributions and providing concluding remarks.

9.2 Contributions

The main aim of this project is to mechanise ontology evolution in Physics by designing novel techniques and integrating them into a system, GALILEO. In Chapter 1, the key hypothesis of the thesis is stated as:

> *A few generic, ontology repair plans can account for a large number of historical instances of ontology evolution in the Physics domain.*

Additionally, the subordinate hypothesis of the thesis is stated as:

> *A few heuristics enable: (i) substantial control over the size of the search in the space containing solution candidates, which is otherwise unmanageable, and (ii) preservation of only physically relevant solutions.*

We claim that both the key and subordinate hypotheses have been verified through the fulfilment of the following five claims:

- *To provide highly general formalisations of ontology repair plans which are designed to resolve ontological faults in Physics ontologies.* Chapter 4 presents formalisations of five ORPs: WMS, Reidealisation, Inconstancy, Unite and Spectrum. The design and formalisations of each ORP are highly general and concise, capable of handling both homogeneous and heterogeneous ontologies, i.e. ontologies that share a common signature and those that use distinct signature symbols.

- *To mechanise ontology evolution by integrating* ORPs *within an environment that provides capabilities to perform reasoning within individual ontologies and over multiple ontologies.* Chapters 5, 6 and 7 discuss our novel approach to realising ontology evolution by integrating ORPs within Isabelle. Thus, GALILEO can be seen as an extension to Isabelle, which essentially equips Isabelle with the functionality of diagnosing ontological faults and repairing conflicting ontologies.

- *To formalise a wide range of case studies from the domain of Physics that are examples of ontology evolution.* Chapters 4 and 8 provide models of a set of case studies in Physics. From our knowledge and experiences, there is no ontology of Physics that is sufficiently rich or adequate for our application. Thus, we have formalised each of the case studies in order to facilitate the evaluation of GALILEO. We have devoted effort to faithfully capture in our models the essential Physics knowledge underlying each case study. The key reason is that we prefer to work with non-trivial theorems, which are inferred from relatively fundamental principles and laws, rather than making trivial inferences, e.g., ones that need only one inference step.

- *To evaluate these techniques over a diverse range of historical records of ontology evolution in Physics.* Also in Chapter 8, we have applied a small but diverse collection of case studies to the ORPs, ranging from school-level Physics, e.g., Boyle's Law, to advanced Astronomy, e.g., Dark Matter. We have shown that the historical solution from each case study can be successfully emulated by GALILEO.

- *To design heuristics that effectively prune the search space containing logically valid diagnoses to become one of a manageable size containing only physically* *meaningful solutions.* Also in Chapters 7 and 8, we have presented details of the four heuristics used for filtering out physically meaningless diagnoses. These heuristics are particularly important, given the typically huge number of unifiers returned by higher-order unification. We have shown that these heuristics can substantially reduce the size of the search space and the remaining solutions represent plausible alternative theories, including the historically correct one.

9.3 Additional Contributions

The focus of this project has been on the study and investigation of a practical approach to the mechanisation of ontology evolution in Physics. Much attention has been placed on the realisation of ontological conflict diagnosis and effecting ontology repair. However, we believe that there are additional contributions related to GALILEO which are meaningful to the research in this field.

We have further contributed to this field in the following ways:

- *Implementing ontologies as locales.* We have viewed ontologies as contexts, so locales are an appropriate machinery for encoding ontologies within Isabelle. Locales let us configure ontologies in a modular way, essentially allowing local scopes to be defined within a working environment. It, therefore, becomes natural to reason across multiple locales.

- *Extending a higher-order theorem prover with methods to perform ontology evolution.* Theorem provers are not designed for ontology evolution, so our work has highlighted an unusual use of Isabelle. For instance, as discussed in Chapter 6, GALILEO's diagnosis component requires a more sophisticated kind of polymorphism in schematic variables than that naturally supported in Isabelle. This is not a shortcoming of the logic itself, but rather a decision the developers made for practical reasons. Schematic type variables do not need to be transformed into functions for most theorem proving applications, but ontology evolution has turned out to be much more demanding in this respect.

- *Utilising bridging axioms for evolving heterogeneous ontologies.* A flexible, generic approach to dealing with ontological faults should be able to handle heterogeneity in languages. Our notion and use of bridging axioms, which are based on the notion of lifting axioms, assist reasoning across heterogeneous ontologies. Because ORPs are intended to be highly general, they are capable of even repairing bridging axioms whenever appropriate.

9.4 Further Work

Throughout the thesis, we have suggested various areas of further work. We outline possible directions for improving and extending the system and its techniques. These range from increasing the degree of automation in the evolution process to applying GALILEO in fresh domains.

9.4.1 Further Understanding of Ontology Evolution in Physics

The focus of this project is to investigate novel techniques to automate ontology evolution in Physics. Complete automation has not been achieved due to intrinsic problems of object-level reasoning in HOL, e.g., computational intractability. However, the techniques developed and evaluated have been shown to be highly general and capable of emulating various historical episodes of ontology evolution in Physics and producing physically plausible theories that give hints of possibly new Physics.

With the foundational understanding of the requirements and the effects of each ORP that we have developed in this project, however, we believe GALILEO and its repair techniques can be improved in several dimensions.

9.4.1.1 Further Increase in Automation

The process of ontology evolution in GALILEO is not completely automated, but all meta-level operations are. User interactions are required to guide the search for a proof of theorems in the object-level ontologies. This is a limitation of reasoning in a highly expressive logic such as HOL rather than weaker logics such as DL. However, we believe the degree of automation in the whole process can be increased with a better understanding of evolving Physics knowledge.

Generally, the proofs of most of the theorems used to trigger ORPs are relatively simple, e.g., rewriting using certain facts, invoking certain simplifiers, etc. We believe tactics, which are functions that combine lower inference rules, could be implemented to reduce the amount of interaction required for reasoning with object-level ontologies.

9.4.1.2 Theoretical Issues

A formal theory specifying the desirable properties that are relevant to ontology repair can help the development of repair mechanisms, e.g., improve robustness and soundness. Like in belief revision, notions such as *minimal repair* and *rationality* may be important to ensuring that an ORP exhibits acceptable behaviours. The formulation could perhaps be expressed as postulates, similar to AGM postulates for belief revision. However, our work is considerably more complex and sophisticated than belief revision, as we make changes to both signatures and axioms. We can foresee that enforcing a notion based on conservative extensions could be much too strong for our application; for instance, Unite has been shown to perform powerful repairs, but the repaired ontologies are not conservative extensions of the old. This is entirely intended, because the purpose of Unite is to enhance the ontology by inserting an axiom that equates the two concepts in question. Similar remarks can be made for most other ORPS, including WMS and Inconstancy.

9.4.2 Applying GALILEO to Other Domains

Each of the ORPs has been designed with a unique kind of fault in mind. Some of these are relatively more relevant to scenarios encountered in scientific domains, whilst some are reminiscent of generic episodes of ontology evolution. Note that other domains may not involve higher-order concepts similar to those in Physics, so they may only need a weaker object-logic.

9.4.2.1 Within Science

The most sophisticated ORP, we believe, is Inconstancy; it specifically looks for an unexpected variation in the value of some property, given several sensory ontologies that together imply such a variation. We argue that the requirement of supplying at least two sensory ontologies is reasonable in the scientific domain. Scientific experiments typically mine data from or analyse collections of observations, measurements or collected data, e.g., surveys in Social Science, so two or more sensory ontologies are generally available at hand. Moreover, hidden dependencies are also commonplace in scientific researches and studies outside Physics. For instance, in 2008, Canadian health researchers suggested that spending money on others and charity promotes happiness (Dunn et al., 2008), so they essentially introduced a new dependency to the degree of happiness. In a Social Science research, income inequality has been recently found to affect social ills and social mobility in societies of the developed world (Wilkinson and Pickett, 2009). We believe there are also various kinds of faults which occur within other areas of Science and are relevant to other ORPs.

9.4.2.2 Beyond Science

We can imagine many scenarios in which ORPs such as Reidealisation and Spectrum are applicable outwith Science. For instance, assigning a concept to a wrong type is a commonly occurring mistake by modellers. Also in programming, variables are often declared using a wrong type in a statically typed language even by experienced programmers. Reidealisation can potentially deal with exactly this kind of problems and produce with a correct typing for the concept in question.

Structures used for the representation of knowledge can often be improved or optimised for a variety of reasons; for instance, an inefficient structure could be an impediment to the understanding of the knowledge being represented. Spectrums, charts and other classifications help display knowledge in a visual way. The Spectrum ORP works with only one ontology, so the input ontology requirement is relatively relax and does not resolve a logical contradiction, but instead enhances the ontology with additional logical power.

9.4.3 Experimenting with Other Logics

Higher-order logic is often regarded to be too expressive for many application domains, especially where achieving a high degree of automation is an important goal. It would therefore be interesting to apply our techniques for ontology evolution to weaker logics, e.g., DL or FOL.

The encodings of all of the ORPs inherently require higher-order logic, as one of the goals of this project is to obtain high generality in each ORP by incorporating higher-order features, e.g., polymorphism and ranging over functions. So, we believe the logic used for the reformulations of the ORPs themselves must be at least second-order. However, the object-level ontologies can be encoded in DL or FOL, as the ORPs are sufficiently expressive to evolve ontologies when the object-level logic is or weaker than HOL.

A weaker logic for the object-level ontologies can potentially increase the degree of automation in the whole ontology evolution process. Most of the interactions required in using GALILEO are used to guide the search for a proof of a theorem in the object-level ontologies. Automated FOL theorem provers have increasingly shown successes in solving hard problems, even in an undecidable logic (Riazanov and Voronkov, 2002; Weidenbach et al., 2009). Indeed, automated reasoning over DL is widely achieved with DL reasoners.

9.4.4 New Ontology Repair Plans

The ORPs presented in this thesis have been shown to emulate a wide ranging variety of episodes of ontology evolution in Physics. That said, most of the case studies identified are suitable specifically to WMS and we believe new ORPs may yield a diminishing return. Nonetheless, the design and development of new ORPs could help us better understand the general ontology evolution process.

During our testing and development, we have discovered examples of conflicts that are not applicable to any of our existing ORPs. One such example is based on the bouncing-ball paradox (§8.2.1.1, p.8.2.1.1). Beside the paradox formulated in §8.2.1.1, another paradox could arise by applying (8.21, p.127) in the right-to-left direction to speak about the values of *Pos* when given values of height, velocity, etc. So,

$$\begin{aligned} Pred \oplus O_b \ &\vdash \ Pos(Ball, Photo(End(Drop))) > 0 \ \vee \qquad (9.1)\\ &\quad\ Pos(Ball, Photo(End(Drop) - \delta)) - \\ &\quad\ Pos(Ball, Photo(End(Drop))) > 0 \\ Obs \ &\vdash \ Pos(Ball, Photo(End(Drop))) = 0 \ \wedge \qquad (9.2)\\ &\quad\ Pos(Ball, Photo(End(Drop) - \delta)) - \\ &\quad\ Pos(Ball, Photo(End(Drop))) = 0 \end{aligned}$$

also hold. The sentences inferred here, (9.1) and (9.2), have a very different physical meaning compared to (8.23 - 8.24, p.127), as O_1 and O_2 are now instantiated to $Pred \oplus O_b$ and *Obs*, respectively. The contradiction inferred in (8.23) and (8.24) comes from a disagreement on the amount of total energy in the ball at the end of the drop, whereas the contradiction inferred in (9.1) and (9.2) comes from a disagreement between the predicted and observed positions of the ball in photos. The final amount of total energy can be inferred to be greater than zero using the predictive theory, but neither the final height nor the final velocity of the ball can be inferred in *Pred* – either or both of these could be positive. Thus, one cannot reason specifically about the values that the height or the velocity of the ball can take, but instead deduce a more complex formula expressing that the position of the ball in the photo taken at the end of the drop is greater than zero *or* the difference between the positions of the ball in two adjacent photos is greater than zero. This, consequently, fails to trigger WMS because (9.1) and (9.2) cannot be matched with the trigger formulae. The WMS repair plan is designed to identify a particular type of discrepancy between the values returned by an application of some function in two ontologies, where the equations derived must be specified as *intervals*. If the two intervals are disjoint, then it is clear that one must be ordered greater than the other. In (9.1) and (9.2), neither the position of the ball in the photo nor the difference in the positions in the two adjacent photos can be inferred in $Pred \oplus O_b$, so the equations cannot be specified as intervals. Therefore, although (9.1) and (9.2) give rise to a contradiction, the failure to trigger WMS is entirely expected. Thus, the conflict described can be considered as a potential avenue for a new ORP.

9.5 Summary

Our experimental results support the hypotheses, showing that ORPs are capable of resolving a variety of ontological faults – both over- and under-specified ontologies. Certain ORPs are more sophisticated and particularly relevant to ontology evolution in scientific domains, e.g., Inconstancy, whereas some others are more generic, e.g., Reidealisation. We believe further automation of ontology evolution will likely require a better fundamental understanding of the ontology evolution process itself, including theoretical and, even, epistemological issues. GALILEO currently demonstrates the applicability of the theory to the Physics domain, but its applicability to other domains is not yet fully explored. That said, given that the evolution techniques that have been integrated in GALILEO are highly generic and expressed in HOL, the same principles underlying the diagnosis and repair components can, in principle, be feasibly applied to ontologies expressed in weaker, more tractable logics. The current implementation of GALILEO could potentially be improved and be more widely applicable in several ways.

Appendix A

Supplementary Diagnosis Results

$$?f \mapsto Approx\,, \qquad ?stuff \mapsto TravelTime\ (LightFrom\ Io)\ Io\ Earth \tag{A.1}$$

$$?f \mapsto \lambda a.\ a\ Earth\ Io\,, \qquad ?stuff \mapsto \lambda a, b.\ Approx\ (TravelTime\ (LightFrom\ b)\ b\ a) \tag{A.2}$$

$$?f \mapsto \lambda a.\ a\ Earth\ Io\,, \qquad ?stuff \mapsto \lambda a, b.\ Approx\ (TravelTime\ (LightFrom\ b)\ Io\ a) \tag{A.3}$$

$$?f \mapsto \lambda a.\ a\ Earth\ Io\,, \qquad ?stuff \mapsto \lambda a, b.\ Approx\ (TravelTime\ (LightFrom\ Io)\ b\ a) \tag{A.4}$$

$$?f \mapsto \lambda a.\ a\ Earth\ (LightFrom\ Io)\,, \qquad ?stuff \mapsto \lambda a, b.\ Approx\ (TravelTime\ b\ Io\ a) \tag{A.5}$$

$$?f \mapsto \lambda a.\ a\ Earth\,, \qquad ?stuff \mapsto \lambda a.\ Approx\ (TravelTime\ (LightFrom\ Io)\ Io\ a) \tag{A.6}$$

$$?f \mapsto \lambda a.\ a\ Io\ Earth\,, \qquad ?stuff \mapsto \lambda a, b.\ Approx\ (TravelTime\ (LightFrom\ a)\ a\ b) \tag{A.7}$$

$$?f \mapsto \lambda a.\ a\ Io\ Earth\,, \qquad ?stuff \mapsto \lambda a, b.\ Approx\ (TravelTime\ (LightFrom\ a)\ Io\ b) \tag{A.8}$$

Figure A.1: Instantiations of $?f$ and $?stuff$ for the Travel Time of Light case study (§8.2.2.1).

$$?f \mapsto \lambda a.\ a\ Io\ Earth\ , \qquad (A.9)$$
$$?stuff \mapsto \lambda a, b.\ Approx\ (TravelTime\ (LightFrom\ Io)\ a\ b)$$
$$?f \mapsto \lambda a.\ a\ Io\ Io\ , \qquad (A.10)$$
$$?stuff \mapsto \lambda a, b.\ Approx\ (TravelTime\ (LightFrom\ a)\ b\ Earth)$$
$$?f \mapsto \lambda a.\ a\ Io\ Io\ , \qquad (A.11)$$
$$?stuff \mapsto \lambda a, b.\ Approx\ (TravelTime\ (LightFrom\ b)\ a\ Earth)$$
$$?f \mapsto \lambda a.\ a\ Io\ (LightFrom\ Io)\ , \qquad (A.12)$$
$$?stuff \mapsto \lambda a, b.\ Approx\ (TravelTime\ b\ a\ Earth)$$
$$?f \mapsto \lambda a.\ a\ Io\ , \qquad (A.13)$$
$$?stuff \mapsto \lambda a.\ Approx\ (TravelTime\ (LightFrom\ a)\ a\ Earth)$$
$$?f \mapsto \lambda a.\ a\ Io\ , \qquad (A.14)$$
$$?stuff \mapsto \lambda a.\ Approx\ (TravelTime\ (LightFrom\ a)\ Io\ Earth)$$
$$?f \mapsto \lambda a.\ a\ Io\ , \qquad (A.15)$$
$$?stuff \mapsto \lambda a.\ Approx\ (TravelTime\ (LightFrom\ Io)\ a\ Earth)$$
$$?f \mapsto \lambda a.\ a\ (LightFrom\ Io)\ Earth\ , \qquad (A.16)$$
$$?stuff \mapsto \lambda a, b.\ Approx\ (TravelTime\ a\ Io\ b)$$
$$?f \mapsto \lambda a.\ a\ (LightFrom\ Io)\ Io\ , \qquad (A.17)$$
$$?stuff \mapsto \lambda a, b.\ Approx\ (TravelTime\ a\ b\ Earth)$$
$$?f \mapsto \lambda a.\ a\ (LightFrom\ Io)\ , \qquad (A.18)$$
$$?stuff \mapsto \lambda a.\ Approx\ (TravelTime\ a\ \ io\ Earth)$$
$$?f \mapsto \lambda a.\ Approx\ (a\ Io\ Earth)\ , \qquad (A.19)$$
$$?stuff \mapsto \lambda a, b.\ TravelTime\ (LightFrom\ a)\ a\ b$$
$$?f \mapsto \lambda a.\ Approx\ (a\ Earth\ Io)\ , \qquad (A.20)$$
$$?stuff \mapsto \lambda a, b.\ TravelTime\ (LightFrom\ b)\ Io\ a$$
$$?f \mapsto \lambda a.\ Approx\ (a\ Earth\ Io)\ , \qquad (A.21)$$
$$?stuff \mapsto \lambda a, b.\ TravelTime\ (LightFrom\ Io)\ b\ a$$
$$?f \mapsto \lambda a.\ Approx\ (a\ Earth\ (LightFrom\ Io))\ , \qquad (A.22)$$
$$?stuff \mapsto \lambda a, b.\ TravelTime\ b\ Io\ a$$

Figure A.1: Instantiations of $?f$ and $?stuff$ for the Travel Time of Light case study (§8.2.2.1) (contd.).

$$?f \mapsto \lambda a.\ \mathit{Approx}\ (a\ \mathit{Earth})\ , \qquad ?\mathit{stuff} \mapsto \mathit{TravelTime}\ (\mathit{LightFrom}\ \mathit{Io})\ \mathit{Io} \tag{A.23}$$

$$?f \mapsto \lambda a.\ \mathit{Approx}\ (a\ \mathit{Io}\ \mathit{Earth})\ , \qquad ?\mathit{stuff} \mapsto \lambda a.\ \mathit{TravelTime}\ (\mathit{LightFrom}\ a)\ a \tag{A.24}$$

$$?f \mapsto \lambda a.\ \mathit{Approx}\ (a\ \mathit{Io}\ \mathit{Earth})\ , \qquad ?\mathit{stuff} \mapsto \lambda a.\ \mathit{TravelTime}\ (\mathit{LightFrom}\ a)\ \mathit{Io} \tag{A.25}$$

$$?f \mapsto \lambda a.\ \mathit{Approx}\ (a\ \mathit{Io}\ \mathit{Earth})\ , \qquad ?\mathit{stuff} \mapsto \mathit{TravelTime}\ (\mathit{LightFrom}\ \mathit{Io}) \tag{A.26}$$

$$?f \mapsto \lambda a.\ \mathit{Approx}\ (a\ \mathit{Io}\ \mathit{Io})\ , \qquad ?\mathit{stuff} \mapsto \lambda a, b.\ \mathit{TravelTime}\ (\mathit{LightFrom}\ a)\ b\ \mathit{Earth} \tag{A.27}$$

$$?f \mapsto \lambda a.\ \mathit{Approx}\ (a\ \mathit{Io}\ \mathit{Io})\ , \qquad ?\mathit{stuff} \mapsto \lambda a, b.\ \mathit{TravelTime}\ (\mathit{LightFrom}\ b)\ a\ \mathit{Earth} \tag{A.28}$$

$$?f \mapsto \lambda a.\ \mathit{Approx}\ (a\ \mathit{Io}\ (\mathit{LightFrom}\ \mathit{Io}))\ , \qquad ?\mathit{stuff} \mapsto \lambda a, b.\ \mathit{TravelTime}\ b\ a\ \mathit{Earth} \tag{A.29}$$

$$?f \mapsto \lambda a.\ \mathit{Approx}\ (a\ \mathit{Io})\ , \qquad ?\mathit{stuff} \mapsto \lambda a.\ \mathit{TravelTime}\ (\mathit{LightFrom}\ a)\ a\ \mathit{Earth} \tag{A.30}$$

$$?f \mapsto \lambda a.\ \mathit{Approx}\ (a\ \mathit{Io})\ , \qquad ?\mathit{stuff} \mapsto \lambda a.\ \mathit{TravelTime}\ (\mathit{LightFrom}\ a)\ \mathit{Io}\ \mathit{Earth} \tag{A.31}$$

$$?f \mapsto \lambda a.\ \mathit{Approx}\ (a\ \mathit{Io})\ , \qquad ?\mathit{stuff} \mapsto \lambda a.\ \mathit{TravelTime}\ (\mathit{LightFrom}\ \mathit{Io})\ a\ \mathit{Earth} \tag{A.32}$$

$$?f \mapsto \lambda a.\ \mathit{Approx}\ (a\ (\mathit{LightFrom}\ \mathit{Io})\ \mathit{Earth})\ , \qquad ?\mathit{stuff} \mapsto \lambda a.\ \mathit{TravelTime}\ a\ \mathit{Io} \tag{A.33}$$

$$?f \mapsto \lambda a.\ \mathit{Approx}\ (a\ (\mathit{LightFrom}\ \mathit{Io})\ \mathit{Io})\ , \qquad ?\mathit{stuff} \mapsto \lambda a, b.\ \mathit{TravelTime}\ a\ b\ \mathit{Earth} \tag{A.34}$$

$$?f \mapsto \lambda a.\ \mathit{Approx}\ (a\ (\mathit{LightFrom}\ \mathit{Io}))\ , \qquad ?\mathit{stuff} \mapsto \lambda a.\ \mathit{TravelTime}\ a\ \mathit{Io}\ \mathit{Earth} \tag{A.35}$$

$$?f \mapsto \lambda a.\ \mathit{Approx}\ (\mathit{TravelTime}\ a\ \mathit{Io}\ \mathit{Earth})\ , \qquad ?\mathit{stuff} \mapsto \mathit{LightFrom}\ \mathit{Io} \tag{A.36}$$

Figure A.1: Instantiations of $?f$ and $?\mathit{stuff}$ for the Travel Time of Light case study (§8.2.2.1) (contd.).

$$?f \mapsto \lambda a.\ Approx\ (TravelTime\ (a\ Io)\ Io\ Earth)\ , \tag{A.37}$$
$$?stuff \mapsto LightFrom$$
$$?f \mapsto \lambda a.\ Approx\ (TravelTime\ (LightFrom\ a)\ a\ Earth)\ , \tag{A.38}$$
$$?stuff \mapsto Io$$
$$?f \mapsto \lambda a.\ Approx\ (TravelTime\ (LightFrom\ a)\ Io\ Earth)\ , \tag{A.39}$$
$$?stuff \mapsto Io$$
$$?f \mapsto \lambda a.\ Approx\ (TravelTime\ (LightFrom\ Io)\ a\ Earth)\ , \tag{A.40}$$
$$?stuff \mapsto Io$$
$$?f \mapsto \lambda a.\ Approx\ (TravelTime\ (LightFrom\ Io)\ Io\ a)\ , \tag{A.41}$$
$$?stuff \mapsto Earth$$
$$?f \mapsto \lambda a.\ a\ , \tag{A.42}$$
$$?stuff \mapsto Approx\ (TravelTime\ (LightFrom\ Io)\ Io\ Earth)$$
$$?f \mapsto \lambda a.\ a\ (TravelTime\ (LightFrom\ Io)\ Io\ Earth)\ , \tag{A.43}$$
$$?stuff \mapsto Approx$$

Figure A.1: Instantiations of $?f$ and $?stuff$ for the Travel Time of Light case study (§8.2.2.1) (contd.).

$$?f \mapsto \mathit{Boyle}, \quad ?\mathit{stuff} \mapsto \mathit{Hg}\ (\mathit{Flask}\ \mathit{Desk}\ \mathit{On})\ \mathit{In} \tag{A.44}$$

$$?f \mapsto \lambda a.\ a\ \mathit{Desk}, \quad ?\mathit{stuff} \mapsto \lambda a.\ \mathit{Boyle}\ (\mathit{Hg}\ (\mathit{Flask}\ a\ \mathit{On})\ \mathit{In}) \tag{A.45}$$

$$?f \mapsto \lambda a.\ a\ (\mathit{Flask}\ \mathit{Desk}\ \mathit{On}), \quad ?\mathit{stuff} \mapsto \lambda a.\ \mathit{Boyle}\ (\mathit{Hg}\ a\ \mathit{In}) \tag{A.46}$$

$$?f \mapsto \lambda a.\ a\ (\mathit{Hg}\ (\mathit{Flask}\ \mathit{Desk}\ \mathit{On})\ \mathit{In}), \quad ?\mathit{stuff} \mapsto \mathit{Boyle} \tag{A.47}$$

$$?f \mapsto \lambda a.\ a\ \mathit{In}, \quad ?\mathit{stuff} \mapsto \lambda a.\ \mathit{Boyle}\ (\mathit{Hg}\ (\mathit{Flask}\ \mathit{Desk}\ \mathit{On})\ a) \tag{A.48}$$

$$?f \mapsto \lambda a.\ a\ \mathit{On}, \quad ?\mathit{stuff} \mapsto \lambda a.\ \mathit{Boyle}\ (\mathit{Hg}\ (\mathit{Flask}\ \mathit{Desk}\ a)\ \mathit{In}) \tag{A.49}$$

$$?f \mapsto \lambda a.\ a, \quad ?\mathit{stuff} \mapsto \mathit{Boyle}\ (\mathit{Hg}\ (\mathit{Flask}\ \mathit{Desk}\ \mathit{On})\ \mathit{In}) \tag{A.50}$$

$$?f \mapsto \lambda a\, b.\ a\ b\ \mathit{Desk}, \quad ?\mathit{stuff} \mapsto \lambda a\, b.\ \mathit{Boyle}\ (\mathit{Hg}\ (\mathit{Flask}\ b\ \mathit{On})\ \mathit{In})\ a \tag{A.51}$$

$$?f \mapsto \lambda a\, b.\ a\ b\ (\mathit{Flask}\ \mathit{Desk}\ \mathit{On}), \quad ?\mathit{stuff} \mapsto \lambda a\, b.\ \mathit{Boyle}\ (\mathit{Hg}\ b\ \mathit{In})\ a \tag{A.52}$$

$$?f \mapsto \lambda a\, b.\ a\ b\ \mathit{In}, \quad ?\mathit{stuff} \mapsto \lambda a\, b.\ \mathit{Boyle}\ (\mathit{Hg}\ (\mathit{Flask}\ \mathit{Desk}\ \mathit{On})\ b)\ a \tag{A.53}$$

$$?f \mapsto \lambda a\, b.\ a\ b\ \mathit{On}, \quad ?\mathit{stuff} \mapsto \lambda a\, b.\ \mathit{Boyle}\ (\mathit{Hg}\ (\mathit{Flask}\ \mathit{Desk}\ b)\ \mathit{In})\ a \tag{A.54}$$

$$?f \mapsto \lambda a.\ \mathit{Boyle}\ (a\ \mathit{Desk}\ \mathit{In}), \quad ?\mathit{stuff} \mapsto \lambda a.\ \mathit{Hg}\ (\mathit{Flask}\ a\ \mathit{On}) \tag{A.55}$$

Figure A.2: Instantiations of $?f$ and $?\mathit{stuff}$ for the Gas Laws case study (§8.2.2.2).

$$?f \mapsto \lambda a.\, Boyle\ (a\ Desk\ On), \quad ?stuff \mapsto \lambda a\, b.\, Hg\ (Flask\ a\ b)\ In \tag{A.56}$$

$$?f \mapsto \lambda a.\, Boyle\ (a\ Desk), \quad ?stuff \mapsto \lambda a.\, Hg\ (Flask\ a\ On)\ In \tag{A.57}$$

$$?f \mapsto \lambda a.\, Boyle\ (a\ (Flask\ Desk\ On)\ In), \quad ?stuff \mapsto Hg \tag{A.58}$$

$$?f \mapsto \lambda a.\, Boyle\ (a\ (Flask\ Desk\ On)), \quad ?stuff \mapsto \lambda a.\, Hg\ a\ In \tag{A.59}$$

$$?f \mapsto \lambda a.\, Boyle\ (a\ In\ Desk), \quad ?stuff \mapsto \lambda a\, b.\, Hg\ (Flask\ b\ On)\ a \tag{A.60}$$

$$?f \mapsto \lambda a.\, Boyle\ (a\ In\ On), \quad ?stuff \mapsto \lambda a\, b.\, Hg\ (Flask\ Desk\ b)\ a \tag{A.61}$$

$$?f \mapsto \lambda a.\, Boyle\ (a\ In), \quad ?stuff \mapsto Hg\ (Flask\ Desk\ On) \tag{A.62}$$

$$?f \mapsto \lambda a.\, Boyle\ (a\ On\ Desk), \quad ?stuff \mapsto \lambda a\, b.\, Hg\ (Flask\ b\ a)\ In \tag{A.63}$$

$$?f \mapsto \lambda a.\, Boyle\ (a\ On\ In), \quad ?stuff \mapsto \lambda a.\, Hg\ (Flask\ Desk\ a) \tag{A.64}$$

$$?f \mapsto \lambda a.\, Boyle\ (a\ On), \quad ?stuff \mapsto \lambda a.\, Hg\ (Flask\ Desk\ a)\ In \tag{A.65}$$

$$?f \mapsto \lambda a.\, Boyle\ (Hg\ (a\ Desk)\ In), \quad ?stuff \mapsto \lambda a.\, Flask\ a\ On \tag{A.66}$$

$$?f \mapsto \lambda a.\, Boyle\ (Hg\ (a\ Desk\ On)\ In), \quad ?stuff \mapsto Flask \tag{A.67}$$

Figure A.2: Instantiations of $?f$ and $?stuff$ for the Gas Laws case study (§8.2.2.2) (contd.).

$$?f \mapsto \lambda a.\ \mathit{Boyle}\ (\mathit{Hg}\ a\ \mathit{In}), \tag{A.68}$$
$$?\mathit{stuff} \mapsto \mathit{Flask}\ \mathit{Desk}\ \mathit{On}$$
$$?f \mapsto \lambda a.\ \mathit{Boyle}\ (\mathit{Hg}\ (a\ \mathit{On})\ \mathit{In}), \tag{A.69}$$
$$?\mathit{stuff} \mapsto \mathit{Flask}\ \mathit{Desk}$$
$$?f \mapsto \lambda a.\ \mathit{Boyle}\ (\mathit{Hg}\ (\mathit{Flask}\ a\ \mathit{On})\ \mathit{In}), \tag{A.70}$$
$$?\mathit{stuff} \mapsto \mathit{Desk}$$
$$?f \mapsto \lambda a.\ \mathit{Boyle}\ (\mathit{Hg}\ (\mathit{Flask}\ \mathit{Desk}\ a)\ \mathit{In}), \tag{A.71}$$
$$?\mathit{stuff} \mapsto \mathit{On}$$
$$?f \mapsto \lambda a.\ \mathit{Boyle}\ (\mathit{Hg}\ (\mathit{Flask}\ \mathit{Desk}\ \mathit{On})\ a), \tag{A.72}$$
$$?\mathit{stuff} \mapsto \mathit{In}$$

Figure A.2: Instantiations of $?f$ and $?\mathit{stuff}$ for the Gas Laws case study (§8.2.2.2) (contd.).

Appendix B

Bouncing-Ball Case Study in Isabelle

Presented in this appendix is the full details of Isabelle for diagnosing and repairing the bouncing-ball paradox (§8.2.1.1) with GALILEO.

B.1 A Modular Formalisation

theory *Ball-Lift*
imports *Main*
../Basic-Ext
RealVector
uses (*../basics .ML*)
(*../matching .ML*)
(*../wms .ML*)
(*../repair .ML*)
(*../locale -analysis.ML*)
(*../inconstancy.ML*)
(*../reidealisation .ML*)
(*../spectrum .ML*)
(*../unite .ML*)
(*../evolution.ML*)
begin

use *../basics.ML*
use *../matching.ML*
use *../repair.ML*

```
use ../locale-analysis.ML
use ../wms.ML
use ../inconstancy.ML
use ../unite.ML
use ../reidealisation.ML
use ../spectrum.ML
use ../evolution.ML

typedecl Event
typedecl Obj
type-synonym Energy = real
type-synonym Time = real
type-synonym Photo = Obj

locale ROOT

locale Signature = ROOT +
  fixes startev :: Event ⇒ Time
  and endev :: Event ⇒ Time
  and drp :: Event
  and Δ :: real

locale OtSig =
  Signature startev endev drp δ
  for startev endev drp δ +
  fixes vel :: Obj ⇒ Time ⇒ real
  and height :: Obj ⇒ Time ⇒ real
  and te :: Obj ⇒ Time ⇒ Energy
  and pe :: Obj ⇒ Time ⇒ Energy
  and ke :: Obj ⇒ Time ⇒ Energy
  and mass :: Obj ⇒ real
  and g :: real
  and ball :: Obj

locale Ot =
  OtSig startev endev drp Δ vel height te pe ke mass g ball
  for startev endev drp Δ vel height te pe ke mass g ball +
```

assumes *te-ax*: $te\ b\ t = pe\ b\ t + ke\ b\ t$
and *pe-ax*: $pe\ x\ t = mass\ x{*}g{*}height\ x\ t$
and *ke-ax*: $ke\ x\ t = 0.5{*}mass\ x{*}vel\ x\ t{*}vel\ x\ t$
and *cons-ax*: $te\ S\ t1 = te\ S\ t2$
and *g-ax*: $g > 0$
and *chrono-ax*: $startev\ drp <= endev\ drp$

Module or locale "Os1" represents more signature elements and the facts for the prediction based on Ot:

locale *Os1* =
Ot startev endev drp Δ *vel height te pe ke mass g ball*
for *startev endev drp* Δ *vel height te pe ke mass g ball* +
assumes *vinit-ax*: $vel\ ball\ (startev\ drp) = 0$
and *hinit-ax*: $height\ ball\ (startev\ drp) > 0$
and *mass-ax*: $mass\ ball > 0$
and *delta-ax*: $\Delta > 0$

Module or locale "Os2Sig" represents more signature elements:

locale *Os2Sig* =
Signature startev endev drp Δ
for *startev endev drp* Δ +
fixes *posn* :: $Obj \Rightarrow Photo \Rightarrow real$
and *photoat* :: $Time \Rightarrow Photo$
and *ball* :: *Obj*

Module or locale "Os2" represents the observed facts:

locale *Os2* =
Os2Sig startev endev drp Δ *posn photoat ball*
for *startev endev drp* Δ *posn photoat ball* +
assumes *posn1-ax* : $posn\ ball\ (photoat\ ((endev\ drp) - \Delta)\ ) = 0$
and *posn2-ax* : $posn\ ball\ (photoat\ (endev\ drp)) = 0$
and *delta-ax*: $\Delta > 0$

Module or locale "Ob" maps the signatures of Ot and Os2 by means of bridging axioms

locale *Ob1* =
Os2Sig startev endev drp Δ *posn photoat ball* +
OtSig startev endev drp Δ *vel height te pe ke mass g ball*
for *startev endev drp* Δ *vel height te pe ke mass g posn photoat ball* +

assumes *ax1-ax*: *height p t* = *posn p* (*photoat t*)
and *ax2-ax*: *vel p t* = (*posn p* (*photoat t*) − *posn p* (*photoat* (*t* − Δ))) / Δ

locale *Ob2* =
OtSig startev endev drp Δ *vel height te pe ke mass g ball* +
Os2Sig startev endev drp Δ *posn photoat ball*
for *startev endev drp* Δ *vel height te pe ke mass g posn photoat ball* +
assumes *ax1-ax*: *height p t* = *posn p* (*photoat t*)
and *ax2-ax*: (*posn p* (*photoat t*) − *posn p* (*photoat* (*t* − Δ))) = *vel p t* ∗ Δ

lemma (**in** *Os1*) *lem1*: *te ball* (*endev drp*) > *0*
proof −
have *ke ball* (*startev drp*) = *0.5* ∗ *mass ball* ∗ *vel ball* (*startev drp*) ∗ *vel ball* (*startev drp*)
using *ke-ax*
by *blast*
with *vinit-ax pe-ax ke-ax*
have ∗: *te ball* (*startev drp*) = *pe ball* (*startev drp*)
by (*metis add-numeral-0 -right mult-zero-left number-of-Pls*
real-mult-commute te-ax)
with *hinit-ax pe-ax mass-ax g-ax*
have ∗∗: *pe ball* (*startev drp*) > *0*
by (*simp add*: *mult-pos-pos*)
from ∗ ∗∗ *cons-ax*
show *?thesis*
by *metis*
qed

locale *Os2Ob2Ot* =
os2: *Os2 startev endev drp* Δ *posn photoat ball* +
ob2: *Ob2 startev endev drp* Δ *vel height te pe ke mass g posn photoat ball* +
ot: *Ot startev endev drp* Δ *vel height te pe ke mass g ball*
for *startev endev drp* Δ *vel height te pe ke mass g posn photoat ball*

lemma (**in** *Os2Ob2Ot*) *singletonsys*: *ALL t*. *te ball t* = *pe ball t* + *ke ball t*
using *te-ax*
by *simp*

lemma (**in** *Os2Ob2Ot*) *lem1*: *te ball* (*endev drp*) = *0*
proof −
  have ∗: *height ball* (*endev drp*) = *0*
    using *ax1-ax posn2-ax*
    by *auto*
  have (*posn ball* (*photoat* (*endev drp*))) − (*posn ball* (*photoat* (*endev drp* − Δ))) = *0*
    using *posn1-ax posn2-ax*
    by *auto*
  hence ∗∗: *vel ball* (*endev drp*) = *0*
    using *ax2-ax delta-ax*
    by *auto*
  then show *?thesis*
    using *singletonsys* ∗ ∗∗ *pe-ax ke-ax*
    by *auto*
qed

Invoke the WMS ORP

wms

Initialise relevant parameters

try-o1 *Os1 startev endev drp* Δ *vel height te pe ke mass g ball*
try-o2 *Os2Ob2Ot startev endev drp* Δ *vel height te pe ke mass g posn photoat ball*
try-fstuff *te ball* (*endev drp*)

Generate proof obligation

verify

Discharge proof obligation

apply (*intro exI* [**where** *x=0*])
using *Os1.lem1 Os2Ob2Ot.lem1*
by (*metis approxeq lessapprox*)

Produce diagnosis

diagnose

Create repaired ontologies

repair-select *8*

end

B.2 A Flat Formalisation Using Factorisation Script

theory *Ball-Lift-Sub*
imports *Main*
  ../Basic-Ext
  RealVector
uses (*../basics .ML*)
    (*../matching .ML*)
    (*../wms .ML*)
    (*../repair .ML*)
    (*../locale -analysis.ML*)
    (*../inconstancy.ML*)
    (*../reidealisation .ML*)
    (*../unite .ML*)
    (*../spectrum .ML*)
    (*../evolution.ML*)
begin
use *../basics.ML*
use *../matching.ML*
use *../repair.ML*
use *../locale-analysis.ML*
use *../wms.ML*
use *../inconstancy.ML*
use *../unite.ML*
use *../reidealisation.ML*
use *../spectrum.ML*
use *../evolution.ML*

typedecl *Event*
typedecl *Obj*
type-synonym *Energy* = *real*
type-synonym *Time* = *real*
type-synonym *System* = *Obj set*
type-synonym *Photo* = *Obj*

locale *ROOT*

locale *Signature* = *ROOT* +
fixes *startev* :: *Event* ⇒ *Time*
and *endev* :: *Event* ⇒ *Time*
and *drp* :: *Event*
and Δ :: *real*

locale *OtSig* =
Signature startev endev drp Δ
for *startev endev drp* Δ +
fixes *vel* :: *Obj* ⇒ *Time* ⇒ *real*
and *height* :: *Obj* ⇒ *Time* ⇒ *real*
and *te* :: *System* ⇒ *Time* ⇒ *Energy*
and *pe* :: *Obj* ⇒ *Time* ⇒ *Energy*
and *ke* :: *Obj* ⇒ *Time* ⇒ *Energy*
and *mass* :: *Obj* ⇒ *real*
and *g* :: *real*
and *ball* :: *Obj*
and *sysball* :: *System*

[command]

python MakeSubontologies.py

[input]

name: locale O1

decl: OtSig startev endev drp Delta vel height te pe ke mass g ball sysball

params: for startev endev drp b vel height te pe ke mass g ball sysball

ax1: "te S t = (Sum x:S. pe x t + ke x t)"

ax2: "pe x t = mass x*g*height x t"

ax3: "ke x t = 0.5*mass x*vel x t*vel x t"

ax4: "te S t1 = te S t2"

ax5: "g > 0"

ax6: "startev drp <= endev drp"

ax7: "vel ball (startev drp) = 0"

ax8: "height ball (startev drp) > 0"

ax9: "mass ball > 0"

ax10: "sysball = ball"

ax11: "Delta > 0"

Some example locales of the 2,046 locales generated by Factorisation script:

locale *O1-203 8* =
OtSig startev endev drp Δ *vel height te pe ke mass g ball sysball*
for *startev endev drp* Δ *vel height te pe ke mass g ball sysball* +
assumes *ax1*: $te\ S\ t = (\sum x \in S.\ pe\ x\ t + ke\ x\ t)$
and *ax2*: $pe\ x\ t = mass\ x * g * height\ x\ t$
and *ax3*: $ke\ x\ t = 0.5 * mass\ x * vel\ x\ t * vel\ x\ t$
and *ax4*: $te\ S\ t1 = te\ S\ t2$
and *ax5*: $g > 0$
and *ax6*: $startev\ drp <= endev\ drp$
and *ax7*: $vel\ ball\ (startev\ drp) = 0$
and *ax9*: $mass\ ball > 0$
and *ax10*: $sysball = \{ball\}$
and *ax11*: $\Delta > 0$

locale *O1-203 9* =
OtSig startev endev drp Δ *vel height te pe ke mass g ball sysball*
for *startev endev drp* Δ *vel height te pe ke mass g ball sysball* +
assumes *ax1*: $te\ S\ t = (\sum x \in S.\ pe\ x\ t + ke\ x\ t)$
and *ax2*: $pe\ x\ t = mass\ x * g * height\ x\ t$
and *ax3*: $ke\ x\ t = 0.5 * mass\ x * vel\ x\ t * vel\ x\ t$
and *ax4*: $te\ S\ t1 = te\ S\ t2$
and *ax5*: $g > 0$
and *ax6*: $startev\ drp <= endev\ drp$
and *ax8*: $height\ ball\ (startev\ drp) > 0$
and *ax9*: $mass\ ball > 0$
and *ax10*: $sysball = \{ball\}$
and *ax11*: $\Delta > 0$

locale *O1-204 0* =
OtSig startev endev drp Δ *vel height te pe ke mass g ball sysball*
for *startev endev drp* Δ *vel height te pe ke mass g ball sysball* +
assumes *ax1*: $te\ S\ t = (\sum x \in S.\ pe\ x\ t + ke\ x\ t)$
and *ax2*: $pe\ x\ t = mass\ x * g * height\ x\ t$
and *ax3*: $ke\ x\ t = 0.5 * mass\ x * vel\ x\ t * vel\ x\ t$
and *ax4*: $te\ S\ t1 = te\ S\ t2$
and *ax5*: $g > 0$
and *ax7*: $vel\ ball\ (startev\ drp) = 0$
and *ax8*: $height\ ball\ (startev\ drp) > 0$
and *ax9*: $mass\ ball > 0$
and *ax10*: $sysball = \{ball\}$
and *ax11*: $\Delta > 0$

locale *O1-204 1* =
OtSig startev endev drp Δ *vel height te pe ke mass g ball sysball*
for *startev endev drp* Δ *vel height te pe ke mass g ball sysball* +
assumes *ax1*: $te\ S\ t = (\sum x \in S.\ pe\ x\ t + ke\ x\ t)$
and *ax2*: $pe\ x\ t = mass\ x * g * height\ x\ t$
and *ax3*: $ke\ x\ t = 0.5 * mass\ x * vel\ x\ t * vel\ x\ t$
and *ax4*: $te\ S\ t1 = te\ S\ t2$
and *ax6*: $startev\ drp <= endev\ drp$
and *ax7*: $vel\ ball\ (startev\ drp) = 0$
and *ax8*: $height\ ball\ (startev\ drp) > 0$
and *ax9*: $mass\ ball > 0$
and *ax10*: $sysball = \{ball\}$
and *ax11*: $\Delta > 0$

locale *O1-204 2* =
OtSig startev endev drp Δ *vel height te pe ke mass g ball sysball*
for *startev endev drp* Δ *vel height te pe ke mass g ball sysball* +
assumes *ax1*: $te\ S\ t = (\sum x \in S.\ pe\ x\ t + ke\ x\ t)$
and *ax2*: $pe\ x\ t = mass\ x * g * height\ x\ t$
and *ax3*: $ke\ x\ t = 0.5 * mass\ x * vel\ x\ t * vel\ x\ t$
and *ax5*: $g > 0$

and *ax6*: *startev drp* <= *endev drp*
and *ax7*: *vel ball* (*startev drp*) = *0*
and *ax8*: *height ball* (*startev drp*) > *0*
and *ax9*: *mass ball* > *0*
and *ax10*: *sysball* = {*ball*}
and *ax11*: Δ > *0*

locale *O1-204 3* =
OtSig startev endev drp Δ *vel height te pe ke mass g ball sysball*
for *startev endev drp* Δ *vel height te pe ke mass g ball sysball* +
assumes *ax1*: *te S t* = (∑ *x*∈*S*. *pe x t* + *ke x t*)
and *ax2*: *pe x t* = *mass x*∗*g*∗*height x t*
and *ax4*: *te S t1* = *te S t2*
and *ax5*: *g* > *0*
and *ax6*: *startev drp* <= *endev drp*
and *ax7*: *vel ball* (*startev drp*) = *0*
and *ax8*: *height ball* (*startev drp*) > *0*
and *ax9*: *mass ball* > *0*
and *ax10*: *sysball* = {*ball*}
and *ax11*: Δ > *0*

locale *O1-204 4* =
OtSig startev endev drp Δ *vel height te pe ke mass g ball sysball*
for *startev endev drp* Δ *vel height te pe ke mass g ball sysball* +
assumes *ax1*: *te S t* = (∑ *x*∈*S*. *pe x t* + *ke x t*)
and *ax3*: *ke x t* = *0.5*∗*mass x*∗*vel x t*∗*vel x t*
and *ax4*: *te S t1* = *te S t2*
and *ax5*: *g* > *0*
and *ax6*: *startev drp* <= *endev drp*
and *ax7*: *vel ball* (*startev drp*) = *0*
and *ax8*: *height ball* (*startev drp*) > *0*
and *ax9*: *mass ball* > *0*
and *ax10*: *sysball* = {*ball*}
and *ax11*: Δ > *0*

locale *O1-204 5* =
OtSig startev endev drp Δ *vel height te pe ke mass g ball sysball*
for *startev endev drp* Δ *vel height te pe ke mass g ball sysball* +
assumes *ax2*: *pe x t* = *mass x*∗*g*∗*height x t*
and *ax3*: *ke x t* = *0.5*∗*mass x*∗*vel x t*∗*vel x t*
and *ax4*: *te S t1* = *te S t2*
and *ax5*: *g* > *0*
and *ax6*: *startev drp* <= *endev drp*
and *ax7*: *vel ball* (*startev drp*) = *0*
and *ax8*: *height ball* (*startev drp*) > *0*
and *ax9*: *mass ball* > *0*
and *ax10*: *sysball* = {*ball*}
and *ax11*: Δ > *0*

locale *O1-204 6* =
OtSig startev endev drp Δ *vel height te pe ke mass g ball sysball*
for *startev endev drp* Δ *vel height te pe ke mass g ball sysball* +
assumes *ax1*: *te S t* = (∑ *x*∈*S*. *pe x t* + *ke x t*)
and *ax2*: *pe x t* = *mass x*∗*g*∗*height x t*
and *ax3*: *ke x t* = *0.5*∗*mass x*∗*vel x t*∗*vel x t*
and *ax4*: *te S t1* = *te S t2*
and *ax5*: *g* > *0*
and *ax6*: *startev drp* <= *endev drp*
and *ax7*: *vel ball* (*startev drp*) = *0*
and *ax8*: *height ball* (*startev drp*) > *0*
and *ax9*: *mass ball* > *0*
and *ax10*: *sysball* = {*ball*}
and *ax11*: Δ > *0*

locale *Os2Sig* =
Signature startev endev drp Δ
for *startev endev drp* Δ +
fixes *posn* :: *Obj* ⇒ *Photo* ⇒ *real*
and *photoat* :: *Time* ⇒ *Photo*

```
  and ball :: Obj
  and sysball :: System


locale O2 =
  Os2Sig startev endev drp Δ posn photoat ball sysball
  for startev endev drp Δ posn photoat ball sysball +
  assumes ax1: posn ball (photoat ((endev drp) − Δ) ) = 0
  and ax2: posn ball (photoat ((endev drp) − Δ) ) = 0
  and ax3: posn ball (photoat (endev drp)) = 0
  and ax4: sysball = {ball}
  and ax5: Δ > 0


locale Ob1 =
  Os2Sig startev endev drp Δ posn photoat ball sysball +
  OtSig startev endev drp Δ vel height te pe ke mass g ball sysball
  for startev endev drp Δ vel height te pe ke mass g posn photoat ball sysball +
  assumes ax1-ax: height p t = posn p (photoat t)
  and ax2-ax: vel p t = (posn p (photoat t) − posn p (photoat (t − Δ) )) / Δ


locale O2ObO1 =
o2: O2 startev endev drp Δ posn photoat ball sysball +
ob: Ob1 startev endev drp Δ vel height te pe ke mass g posn photoat ball sysball +
o1: O1-204 2 startev endev drp Δ vel height te pe ke mass g ball sysball
for startev endev drp Δ vel height te pe ke mass g posn photoat ball sysball


lemma (in O1-204 6) lem1: te sysball (endev drp) > 0
proof −
  have ke ball (startev drp) = 0.5 ∗ mass ball ∗ vel ball (startev drp) ∗ vel ball (startev drp)
    using ax3
    by blast
  with ax7 ax3 ax2 ax1 ax10
  have ∗: te sysball (startev drp) = pe ball (startev drp)
    by (simp add: mult-right.zero )
```

with *ax8 ax2 ax9 ax5*
have $**$: *pe ball* (*startev drp*) > *0*
by (*simp add*: *mult-pos-pos*)
from $*$ $**$ *ax4*
show *?thesis*
by (*metis real-less-def*)
qed

lemma (**in** *O2ObO1*) *lem1*: *te sysball* (*endev drp*) = *0*
proof −
have $*$: *height ball* (*endev drp*) = *0*
using *ax1-ax o2.ax3*
by *auto*
have (*posn ball* (*photoat* (*endev drp*))) − (*posn ball* (*photoat* (*endev drp* − Δ))) = *0*
using *o2.ax3 o2.ax2*
by *auto*
hence $**$: *vel ball* (*endev drp*) = *0*
using *ax2-ax o2.ax5*
by *auto*
then show *?thesis*
using *ax1 ax10* $*$ $**$ *o1.ax2 o1.ax3*
by *auto*
qed

Invoke the WMS ORP

wms

Initialise relevant parameters

try-o1 *O1-2046 startev endev drp* Δ *vel height te pe ke mass g ball*
try-o2 *O2ObO1 startev endev drp* Δ *vel height te pe ke mass g posn photoat ball*
try-fstuff *te ball* (*endev drp*)

Generate proof obligatoin

verify

Discharge proof obligation

apply (*intro exI* [**where** *x=0*])
using *O1-2046.lem1 Os2Ob2Ot.lem1*
by *simp*

Produce diagnosis

diagnose

Create repaired ontologies

repair-select *8*

end

Bibliography

Alchourrón, C. E., Gärdenfors, P., and Makinson, D. (1985). On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50(2):510–530.

Amgoud, L., Dimopoulos, Y., and Moraitis, P. (2007). A unified and general framework for argumentation-based negotiation. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, page 158. ACM.

Andrews, P., Miller, D., Cohen, E., and Pfenning, F. (1984). Automating higher-order logic. *Automated theorem proving: After*, 25:169–192.

Atkinson, K., Bench-Capon, T., and Mcburney, P. (2005). A dialogue game protocol for multi-agent argument over proposals for action. *Autonomous Agents and Multi-Agent Systems*, 11(2):153–171.

Autexier, S., Hutter, D., Mossakowski, T., and Schairer, A. (2002). The development graph manager MAYA. In *Proceedings of the 9th International Conference on Algebraic Methodology and Software Technology*, pages 495–501. Springer-Verlag London, UK.

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P. F. (2003). The description logic handbook: theory, implementation, and applications.

Baader, F., Peñaloza, R., and Suntisrivaraporn, B. (2007). Pinpointing in the description logic el+\ mathcal {EL}^+. *KI 2007: Advances in Artificial Intelligence*, pages 52–67.

Bailin, S. and Truszkowski, W. (2002). Ontology negotiation between intelligent information agents. *The Knowledge Engineering Review*, 17(01):7–19.

Ballarin, C. (2006). Interpretation of locales in isabelle: Theories and proof contexts. In *Mathematical Knowledge Management*, volume LNCS 4108, pages 31–43. Springer Berlin/Heidelberg.

Baltopoulos, I., Borgström, J., and Gordon, A. (2011). Maintaining database integrity with refinement types. *ECOOP 2011–Object-Oriented Programming*, pages 484–509.

Barker, K., Porter, B., and Clark, P. (2001). A library of generic concepts for composing knowledge bases. In *Proceedings of the 1st international conference on Knowledge capture*, pages 14–21. ACM.

Barwise, J. and Seligman, J. (1997). *Information Flow : The Logic of Distributed Systems*, volume 44 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press.

Bench-Capon, T. and Malcolm, G. (1999). Formalising ontologies and their relations. In *Database and Expert Systems Applications*, pages 815–815. Springer.

Bertot, Y. and Castéran, P. (2004). Coq'art. *by Springer-Verlag*.

Binmore, K. and Vulkan, N. (1999). Applying game theory to automated negotiation. *Netnomics*, 1(1):1–9.

Brickley, D. and Guha, R. (2004). RDF Vocabulary Description Language 1.0: RDF Schema. *W3C Recommendation*.

Bundy, A. (1979). *MECHO: A program to solve mechanics problems*. Edinburgh University.

Bundy, A. (2009). Unite: A new plan for automated ontology evolution in physics. *ARCOE-09*, page 34.

Bundy, A. (2010). The spectrum ontology repair plan. Blue Book Note 1712.

Bundy, A. and Chan, M. (2008). Towards ontology evolution in physics. In Hodges, W. and de Queiroz, R., editors, *Logic, Language, Information and Computation*, volume 5110 of *Lecture Notes in Computer Science*, pages 98–110. Springer Berlin / Heidelberg.

Bundy, A. and McNeill, F. (2006). Representation as a fluent: An AI challenge for the next half century. *IEEE Intelligent Systems*, 21(3):85–87.

Castelfranchi, C. (1995). Guarantees for autonomy in cognitive agent architecture. In *Proceedings of the workshop on agent theories, architectures, and languages*, pages 56–70, NY, USA. Springer-Verlag.

Chan, M. and Bundy, A. (2008). Inconstancy: An ontology repair plan for adding hidden variables. In Bringsjord, S. and Shilliday, A., editors, *Symposium on Automated Scientific Discovery*, number FS-08-03 in Technical Report, pages 10–17. AAAI Press. ISBN 978-1-57735-395-9.

Chan, M. and Bundy, A. (2009). An architecture of galileo: A system for automated ontology evolution in physics. *ARCOE-09*, page 37.

Chan, M., Lehmann, J., and Bundy, A. (2010a). A contextual approach to detection of conflicting ontologies. In *Proceedings of ECAI'10 Workshop on Automated Reasoning about Context and Ontology Evolution*, page 23.

Chan, M., Lehmann, J., and Bundy, A. (2010b). Higher-order representation and reasoning for automated ontology evolution. In *Proceedings of the 2010 International Conference on Knowledge Engineering and Ontology Development*, pages 84–93.

Chan, M., Lehmann, J., and Bundy, A. (2011). Galileo: A system for automating ontology evolution. *ARCOE-11*, page 46.

Chaudhri, V., John, B., Mishra, S., Pacheco, J., Porter, B., and Spaulding, A. (2007). Enabling experts to build knowledge bases from science textbooks. In *Proceedings of the 4th international conference on Knowledge capture*, pages 159–166. ACM.

Church, A. (1940). A formulation of the simple theory of types. *Symbolic Logic*, 5(1):56–68.

Constable, R. L., Allen, S. F., Bromley, H. M., et al. (1986). *Implementing Mathematics with the Nuprl Proof Development System*. Prentice Hall.

Curino, C., Moon, H., and Zaniolo, C. (2008). Graceful database schema evolution: the prism workbench. *Proceedings of the VLDB Endowment*, 1(1):761–772.

Darwiche, A. and Pearl, J. (1997). On the logic of iterated belief revision. *Artificial Intelligence*, 89(1-2):1–29.

Davenport, R. (1976). Database integrity. *The Computer Journal*, 19(2):110.

De Kleer, J. (1977). Multiples representations of knowledge in a mechanics problem-solver. In *Proceedings of the 5th international joint conference on Artificial intelligence-Volume 1*, pages 299–304. Morgan Kaufmann Publishers Inc.

De Kleer, J. and Brown, J. (1984). A qualitative physics based on confluences. *Artificial intelligence*, 24(1-3):7–83.

diSessa, A. (1983). Phenomenology and the evolution of intuition. In Stevens, A. and Gentner, D., editors, *Mental Models*, pages 15–33. Erlbaum.

Doan, A., Madhavan, J., Domingos, P., and Halevy, A. (2004). Ontology matching: A machine learning approach. *Handbook on Ontologies in Information Systems*, pages 397–416.

Doran, P. (2006). Ontology reuse via ontology modularisation. In *KnowledgeWeb PhD Symposium*, volume 2006. Citeseer.

Dowek, G. (1994). Third order matching is decidable. *Annals of Pure and Applied Logic*, 69(2):135–155.

Du, J., Qi, G., and Shen, Y. (2008). Lexicographical inference over inconsistent dl-based ontologies. *Web Reasoning and Rule Systems*, pages 58–73.

Dunn, E., Aknin, L., and Norton, M. (2008). Spending money on others promotes happiness. *Science*, 319(5870):1687–1688.

Dunne, P. and Bench-Capon, T. (2006). Multi-agent agreements about actions through argumentation. *Frontiers in Artificial Intelligence and Applications*, page 323.

Ellis, G. and Uzan, J. (2005). c is the speed of light, isn't it? *American journal of physics*, 73:240.

Faratin, P., Sierra, C., and Jennings, N. (2002). Using similarity criteria to make issue trade-offs in automated negotiations. *artificial Intelligence*, 142(2):205–237.

FermiLab, F. N. A. L. (2012). Fermilab: Holometer. `http://holometer.fnal.gov/`. [Online; accessed 21-Feb-2012].

Fikes, R. and Nilsson, N. (1972). Strips: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3):189–208.

Flouris, G., Manakanatas, D., Kondylakis, H., Plexousakis, D., and Antoniou, G. (2008). Ontology change: classification and survey. *The Knowledge Engineering Review*, 23(02):117–152.

Formica, A. (2008). Concept similarity in formal concept analysis: An information content approach. *Knowledge-Based Systems*, 21(1):80–87.

Freund, M. and Lehmann, D. (2002). Belief Revision and Rational Inference. *Technical Report TR94-16, Hebrew University*.

Gärdenfors, P. and Makinson, D. (1988). Revisions of knowledge systems using epistemic entrenchment. In *Proceedings of the 2nd conference on Theoretical aspects of reasoning about knowledge*, pages 83–95. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

Gärdenfors, P. and Rott, H. (1995). Belief revision. In Gabbay, D., Hogger, C. J., and Robinson, J., editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 4, pages 35–132. Oxford Science Publications.

Ghilardi, S., Lutz, C., and Wolter, F. (2006). Did I damage my ontology? A case for conservative extensions in description logic. *Proceedings of the Tenth International Conference on Principles of Knowledge Representation and Reasoning*, pages 187–197.

Giunchiglia, F. and Shvaiko, P. (2004). Semantic matching. *The Knowledge Engineering Review*, 18(03):265–280.

Giunchiglia, F., Shvaiko, P., and Yatskevich, M. (2004). S-match: an algorithm and an implementation of semantic matching. *The semantic web: research and applications*, pages 61–75.

Giunchiglia, F., Yatskevich, M., and Shvaiko, P. (2007). Semantic matching: Algorithms and implementation. In *Journal on Data Semantics IX*, pages 1–38. Springer-Verlag.

Gligorov, R., ten Kate, W., Aleksovski, Z., and Van Harmelen, F. (2007). Using google distance to weight approximate ontology matches. In *Proceedings of the 16th international conference on World Wide Web*, pages 767–776. ACM.

Gordon, M. J., Milner, A. J., and Wadsworth, C. P. (1979). *Edinburgh LCF - A mechanised logic of computation*, volume 78 of *Lecture Notes in Computer Science*. Springer-Verlag.

Gordon, M. J. and Pitts, A. M. (1994). The hol logic and system. In Bowen, J., editor, *Towards Verified Systems*, chapter 3, pages 49–70. Elsevier Science B. V.

Grau, B., Horrocks, I., Kazakov, Y., and Sattler, U. (2007). Just the right amount: Extracting modules from ontologies. In *Proceedings of the 16th international conference on World Wide Web*, page 726. ACM.

Grau, B. C., Horrocks, I., Kazakov, Y., and Sattler, U. (2001). Extracting modules from ontologies : Theory and practice (technical report). *The University of Manchester Oxford Road Manchester M13 9PL UK February 2007*, pages 1–39.

Guinchiglia, F. and Yatskevich, M. (2004). Element level semantic matching. *Meaning Coordination and Negotiation (MCN-04)*, page 37.

Haase, P. and Qi, G. (2007). An analysis of approaches to resolving inconsistencies in dl-based ontologies. In *Proceedings of the International Workshop on Ontology Dynamics (IWOD'07)*.

Haase, P., van Harmelen, F., Huang, Z., Stuckenschmidt, H., and Sure, Y. (2005). A Framework for Handling Inconsistency in Changing Ontologies. In *The Semantic Web-ISWC 2005: 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland, November 6-10, 2005: Proceedings*. Springer Verlag.

Hayes, P. (1985). Naive physics I: Ontology for liquids. In Hobbs, J. and Moore, R., editors, *Formal theories of the commonsense world*, pages 71–108. Ablex.

Hayes, P. et al. (1978). The naive physics manifesto.

Hendler, J. (2001). Agents and the semantic web. *Intelligent Systems, IEEE*, 16(2):30–37.

Hindley, R. (1969). The principal type-scheme of an object in combinatory logic. *Transactions of the american mathematical society*, 146:29–60.

Huet, G. (1975). A unification algorithm for typed lambda calculus. *Theoretical Computer Science*, 1:27–57.

Huet, G. (1997). The zipper. *Journal of Functional Programming*, 7(5):549–554.

Huet, G. (2002). Higher order unification 30 years later. *Theorem Proving in Higher Order Logics*, pages 241–258.

Huet, G. and Lang, B. (1978). Proving and applying program transformation expressed with second order patterns. *Acta Informatica*, 11:31–55.

Ichise, R. (2008). Machine learning approach for ontology mapping using multiple concept similarity measures. In *Computer and Information Science, 2008. ICIS 08. Seventh IEEE/ACIS International Conference on*, pages 340–346. IEEE.

Jennings, N., Faratin, P., Lomuscio, A., Parsons, S., Wooldridge, M., and Sierra, C. (2001). Automated negotiation: prospects, methods and challenges. *Group Decision and Negotiation*, 10(2):199–215.

Jensen, F. (1996). *An introduction to Bayesian networks*, volume 36. UCL Press London.

Ji, Q., Haase, P., Qi, G., Hitzler, P., and Stadtmüller, S. (2009). RaDON—Repair and Diagnosis in Ontology Networks. *The Semantic Web: Research and Applications*, pages 863–867.

Kakas, A. and Moraitis, P. (2006). Adaptive agent negotiation via argumentation. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 384–391. ACM.

Kalfoglou, Y. and Schorlemmer, M. (2002). Information flow based ontology mapping. In *Proceedings of the 1st International Conference on Ontologies, Databases and Application of Semantics (ODBASE'02), Irvine, CA, USA*, pages 1132–1151.

Kalfoglou, Y. and Schorlemmer, M. (2003a). If-map: An ontology-mapping method based on information-flow theory. *Journal on data semantics I*, pages 98–127.

Kalfoglou, Y. and Schorlemmer, M. (2003b). Ontology mapping: the state of the art. *The Knowledge Engineering Review*, 18(1):1–31.

Kalfoglou, Y. and Schorlemmer, M. (2004). Formal support for representing and automating semantic interoperability. *The semantic web: Research and applications*, pages 45–60.

Kalyanpur, A. (2006). Debugging and repair of OWL ontologies.

Kalyanpur, A., Parsia, B., Cuenca-Grau, B., and Sirin, E. (2006a). Axiom pinpointing: Finding (precise) justifications for arbitrary entailments in OWL-DL. Technical report.

Kalyanpur, A., Parsia, B., Horridge, M., and Sirin, E. (2007). Finding all justifications of OWL DL entailments. *Lecture Notes in Computer Science*, 4825:267.

Kalyanpur, A., Parsia, B., Sirin, E., and Grau, B. (2006b). Repairing unsatisfiable concepts in OWL ontologies. In *ESWC*, pages 170–184. Springer.

Kalyanpur, A., Parsia, B., Sirin, E., Grau, B., and Hendler, J. (2006c). Swoop: A web ontology editing browser. *Web Semantics: Science, Services and Agents on the World Wide Web*, 4(2):144–153.

Katsuno, H. and Mendelzon, A. O. (1991). Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 52(3):263–294.

Kraus, S., Sycara, K., and Evenchik, A. (1998). Reaching agreements through argumentation: a logical model and implementation. *Artificial Intelligence*, 104(1-2):1–69.

Lakatos, I. (1976). *Proofs and Refutations: The Logic of Mathematical Discovery*. Cambridge University Press.

Langley, P. (1980). Descriptive discovery processes: Experiments in baconian science. Technical report, Carnegie-Mellon University.

Langley, P. (1981). Data-driven discovery of physical laws. *Cognitive Science*, 5(1):31–54.

Langley, P. (1998). The Computer-Aided Discovery of Scientific Knowledge. In *Discovery Science: First International Conference, DS'98, Fukuoka, Japan, December 14-16, 1998, Proceedings*. Springer.

Lehmann, J., Bundy, A., and Chan, M. (2011). Evolution of inconsistent ontologies in physics. In *Proceedings of the Symposium on Inconsistency Robustness 2011.*

Lehmann, J., Chan, M., and Bundy, A. (2012). A higher-order approach to ontology evolution in physics. *Journal on Data Semantics*, pages 1–25.

Lenat, D. (1995). CYC: a large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38.

Lenat, D. B. (1977). Automated theory formation in mathematics. In Reddy, R., editor, *Proceedings of IJCAI-77*, pages 833–842. International Joint Conference on Artificial Intelligence.

Madarász, J., Németi, I., and Székely, G. (2006). Twin paradox and the logical foundation of relativity theory. *Foundations of Physics*, 36(5):681–714.

Manola, F., Miller, E., et al. (2004). RDF Primer. *W3C Recommendation*.

McBrien, P. and Poulovassilis, A. (2002). Schema Evolution in Heterogeneous Database Architectures, A Schema Transformation Approach. *Proceedings of the 14th International Conference on Advanced Information Systems Engineering*, pages 484–499.

McCarthy, J. (1980). Circumscription — a form of non-monotonic reasoning. *Artificial Intelligence*, 13:27–39. Also in Readings in Nonmonotonic Reasoning, Ginsberg, M. L. (ed), Morgan Kauffman, 1987.

McCarthy, J. (1986). Applications of circumscription to formalizing common-sense knowledge. *Artificial Intelligence*, 28:89–116. Also in Readings in Nonmonotonic Reasoning, Ginsberg, M. L. (ed), Morgan Kauffman, 1987.

McCarthy, J. and Buvac, S. (1998). Formalizing context (expanded notes). *Computing natural language*, 81:13–50.

McCarthy, J. and Hayes, P. (1969). Some philosophical problems from the standpoint of artificial intelligence. *Machine intelligence*, 4(463-502):288.

McNeill, F. and Bundy, A. (2007). Dynamic, automatic, first-order ontology repair by diagnosis of failed plan execution. *International Journal On Semantic Web and Information Systems*, 3(3):1–35. Special issue on ontology matching.

Milgrom, M. (1983). A modification of the Newtonian dynamics as a possible alternative to the hidden mass hypothesis. *The Astrophysical Journal*, 270:365–370.

Miller, D. and Nadathur, G. (1986). Higher-order logic programming. In *Third International Conference on Logic Programming*, pages 448–462. Springer.

Miller, G. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Milner, R. (1978). A theory of type polymorphism in programming. *Journal of computer and system sciences*, 17(3):348–375.

Nadathur, G. and Miller, D. (1998). Higher-order logic programming. In Gabbay, D. M., Hogger, C. J., and Robinson, J. A., editors, *Handbook of Logics for Artificial Intelligence and Logic Programming*, volume 5, pages 499–590. Clarendon Press, Oxford, England.

Niepert, M., Meilicke, C., and Stuckenschmidt, H. (2010). A probabilistic-logical framework for ontology matching. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*.

Nipkow, T., Paulson, L. C., and Wenzel, M. (2002). *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*, volume 2283 of *LNCS*. Springer.

Osborne, M. and Rubinstein, A. (1994). *A course in game theory*.

Padovani, V. (2000). Decidability of fourth-order matching. *Mathematical Structures in Computer Science*, 10(3):361–372.

Parsons, S. and Wooldridge, M. (2002). Game theory and decision theory in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 5(3):243–254.

Paulson, L. (1986). Natural deduction as higher order resolution. *Journal of Logic Programming*, 3:237–258.

Paulson, L. C. (1994). *Isabelle: A generic theorem prover*. Springer-Verlag.

Pease, A., Colton, S., Smaill, A., and Lee, J. (2004). A model of Lakatos's philosophy of mathematics. In *Proceedings of the Second European Computing and Philosophy Conference, E-CAP2004*, University of Pavia.

Penaloza, R. and Sertkaya, B. (2009). Axiom pinpointing is hard. In *Proceedings of the 2009 International Workshop on Description Logics (DL2009)*, volume 477.

Pfenning, F. and Schürmann, C. (1999). System description: Twelf — a meta-logical framework for deductive systems. In Ganzinger, H., editor, *Proceedings of the 16th International Conference on Automated Deduction (CADE-16)*, number 1632 in LNAI, pages 202–206. Springer-Verlag.

Pietrzykowski, T. (1973). A complete mechanization of second-order type theory. *Journal of the ACM (JACM)*, 20(2):333–364.

Pólya, G. (1945). *How to Solve It.* Princeton University Press.

Rahwan, I., Sonenberg, L., Jennings, N., and McBurney, P. (2007). Stratum: A methodology for designing heuristic agent negotiation strategies. *Applied Artificial Intelligence*, 21(6):489–527.

Reiter, R. (1978). On closed world data bases. In *Logic and Data Bases*, pages 55–76.

Reiter, R. (1979). A logic for default reasoning. Technical Report 79-8, University of British Columbia.

Reiter, R. (1980). A logic for default reasoning. *Artificial Intelligence*, pages 81–132.

Riazanov, A. and Voronkov, A. (2002). The design and implementation of vampire. *AI communications*, 15(2):91–110.

Ribeiro, M. and Wassermann, R. (2009). Base revision for ontology debugging. *Journal of Logic and Computation*, 19(5):721–743.

Robinson, A. and Voronkov, A., editors (2001). *Handbook of Automated Reasoning*. Elsevier. 2 volumes.

Roddick, J. (1992). Schema evolution in database systems: an annotated bibliography. *ACM SIGMOD Record*, 21(4):35–40.

Rosenschein, J. and Zlotkin, G. (1994). *Rules of encounter: designing conventions for automated negotiation among computers*. the MIT Press.

Rowe, J. and Partridge, D. (1993). Creativity: a survey of AI approaches. *Artificial Intelligence Review*, 7(1):43–70.

Rubin, V. C., Thonnard, N., and Ford, W. K., J. (1980). Rotational properties of 21 SC galaxies with a large range of luminosities and radii, from NGC 4605 (R = 4kpc) to UGC 2885 (R = 122 kpc). *Astrophysical Journal*, 238:471.

Sabou, M., d'Aquin, M., and Motta, E. (2008). Exploring the semantic web as background knowledge for ontology matching. *Journal on Data Semantics XI*, pages 156–190.

Schulz, S. (2002). E-a brainiac theorem prover. *AI Communications*, 15(2-3):111–126.

Shadbolt, N., Berners-Lee, T., and Hall, W. (2006). The semantic web revisited. *IEEE Intelligent Systems*, 21(3):96–101.

Stearns, M., Price, C., Spackman, K., and Wang, A. (2001). Snomed clinical terms: overview of the development process and project status. In *Proceedings of the AMIA Symposium*, page 662. American Medical Informatics Association.

Stirling, C. (2009). Decidability of higher-order matching. *Logical Methods in Computer Science*, 5(3):2.

Stojanovic, L., Maedche, A., Motik, B., and Stojanovic, N. (2002). User-Driven Ontology Evolution Management. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, pages 285–300. Springer-Verlag London, UK.

Strachey, C. (2000). Fundamental concepts in programming languages. *Higher-order and symbolic computation*, 13(1):11–49.

Székely, G. (2010). First-order logic investigation of relativity theory with an emphasis on accelerated observers. *Arxiv preprint arXiv:1005.0973*.

Tamma, V., Wooldridge, M., Blacoe, I., and Dickinson, I. (2002). An ontology based approach to automated negotiation. *Agent-Mediated Electronic Commerce IV. Designing Mechanisms and Systems*, pages 317–334.

Van Brakel, J. (1992). The complete description of the frame problem. *Psycoloquy*, 3(60).

Ventrone, V. (1991). Semantic heterogeneity as a result of domain evolution. *ACM SIGMOD Record*, 20(4):16–20.

W3C (2012). OWL 2 Web Ontology Language Document Overview.

Weidenbach, C., Dimova, D., Fietzke, A., Kumar, R., Suda, M., and Wischnewski, P. (2009). Spass version 3.5. *Automated Deduction–CADE-22*, pages 140–145.

Wenzel, M. (1997). Type classes and overloading in higher-order logic. In *Theorem Proving in Higher Order Logics*, pages 307–322. Springer.

Wenzel, M. (2007). Isabelle/Isar — a generic framework for human-readable proof documents. In Matuszewski, R. and Zalewska, A., editors, *From Insight to Proof — Festschrift in Honour of Andrzej Trybulec*, volume 10(23) of *Studies in Logic, Grammar, and Rhetoric*. University of Bialystok.

Wilkinson, R. and Pickett, K. (2009). *The spirit level*. Bloomsbury Press New York.

Wiser, M. and Carey, S. (1983). When heat and temperature were one. In Stevens, A. and Gentner, D., editors, *Mental Models*, pages 267–297. Erlbaum.

Witbrock, M. (2011). Knowledge for/from people for/from computers. In *Proceedings of the 1st international workshop on Search and mining entity-relationship data*, pages 1–2. ACM.

Wooldridge, M. (2002). *An Introduction to MultiAgent Systems*. John Wiley and Sons.

Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152.

Zablith, F. (2008). Dynamic ontology evolution. *ISWC Doctoral Consortium*.

Zaionc, M. (1985). The set of unifiers in typed λ-calculus as regular expression. In *Rewriting Techniques and Applications*, pages 430–440. Springer.