FORMALIZING ELABORATION TOLERANCE

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Aarati Parmar
August 2003

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

John McCarthy
(Principal Adviser)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Richard Fikes

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Johan van Benthem
(Philosophy)

Approved for the University Committee on Graduate Studies:

Abstract

This thesis examines the concept of *elaboration tolerance*. A representation, whether it be a logical theory, computer program, or even Bayesian network, is elaboration tolerant to the extent it is easy to change to represent new facts. Natural language is the model for elaboration tolerance; in everyday discourse one can easily and succinctly change one's declarations by adding an extra sentence or two.

This thesis has two major contributions. The first is a framework which elucidates how the semantics of a representation affects its elaboration tolerance, and the viability of principles that promote elaboration tolerance.

The second contribution is a thorough examination of what we call *additive elaboration tolerance*. In this paradigm, a representation is modified simply by adding the formula whose meaning corresponds to the desired change. This modality of changing representations is attractive because it avoids "brain surgery" – we do not have to tinker explicitly with the representation at all. We show how to endow a given representation S_1 with additive elaboration tolerance by embedding it in another system $S_{Ab} \cdot S_1 = S_{1,Ab}$. This embedding is intuitive as it models human discourse. It is also sound, preserving facts true in the original representation.

Elaboration tolerance shares concepts with the fields of *belief revision* and *reformulation*. We explore how the AGM postulates relate to our construction $S_{1,Ab}$. We also show how standard problems of nonmonotonic reasoning can be re-framed in terms of searching for elaboration tolerant representations of commonsense theories. We conclude with deeper insights about elaboration tolerance, and its importance to the future of the logical AI program.

Acknowledgments

I could not have come this far without the support of my mom and dad. Their encouragement and dedication to my education have been fundamental to my success. I have been fueled by the immense confidence my younger siblings, Anju and Abhis, have in me. Even our dog Amy contributed, as her enthusiasm for long walks gave me a forum in which to meditate upon my ideas.

My sequence of academic endeavors begins with my teachers. Mrs. Stanton introduced me to my first computer in fifth grade at Episcopal Day School, and I have been happily programming them ever since. Somehow Mrs. Kim, taught me the formidable subject algebra without me even realizing it, thinking it was only "Advanced Math." My Master's thesis advisors at MIT, Helen Meng and Stephanie Seneff, sparked my passion for research and solving problems. They are the reasons I came to pursue a doctorate in Computer Science at Stanford University.

At Stanford, my first run-in with my advisor showed me the importance of logic. In the Spring of 1997, John McCarthy, father of artificial intelligence, described to me his recent formalization of speech acts, Elephant 2000. I was amazed to learn that Elephant was not programmed in any particular language (not even LISP!), but consisted entirely of a formalization in logic. As I have realized since then, this logical approach makes the *essence* of the behavior explicit through formal sentences, rather than only approximating it through some computer program. This observation led me to my exciting journey into logic and higher mathematics.

This journey was lit by three main beacons. Solomon Feferman introduced me to the fundamentals: logic, model theory, recursion theory, and set theory. Grisha Mints took me on a engaging tour of what can be done with logic, such as formalizing choice operators and using intuitionistic logics. Johan van Benthem took the stark, impassive mathematical concepts of modal logics and turned them into fanciful games and puzzles over which it was (is) a joy to ponder.

But my lighthouse is my advisor, John McCarthy. Ever since I wrote my first set of computer programs I wanted to find out how to make them smarter. I knew there was a methodology for doing so, but I did not know what it was till I met John. I am lucky to have such an insightful, intellectually stimulating advisor, with such an elegant, principled approach for building intelligent robots. The bonus is that John knows a lot of funny jokes and amusing anecdotes.

My friends and colleagues here at Stanford made this trip worthwhile, from my fellow students Greg, Liadan, Neil, Karen, Karl, Diane, ..., to my special AI posse Tom, Eyal, Lise, Xavier, Pedrito, Ursula, Simon, Uri, and Gleb. Special thanks go to Guha, Patrick, Satish, Kelly, Rob, and Mike S. I am also grateful for the guidance of my colleagues Daphne, Richard, Ed, Sheila, Vaughan, Mike, Berthe, Paulo, and Selene. Also, my final year would not have been nearly as fun without my officemates Neil and Joey, and our honorary one, Ron. Never again shall I witness such amazing simulations of bouncing rubber bricks and deflating statues.

I want to also acknowledge the members of my reading committee for their patience with this work. I am particularly grateful for the time they spent carefully reading this thesis and helping me make it better.

Finally, this thesis would not have been possible without the guidance of my "doctoral dissertation manager," Paul. Paul's faith is what powered the last leg of this long journey, and it is only because of him that this product was "shipped out the door," on time. I owe him a great deal, the magnitude of which (ironically) cannot be properly represented in any declarative formalism.

This research has been partly supported by SNWSC contract N66001-00-C-8018 as well as a National Science Foundation Graduate Research Fellowship.

Contents

List of Figures

Part I

Introduction to Elaboration Tolerance

Chapter 1

Framework and Motivation

The notion of *elaboration tolerance* is originally introduced in John McCarthy's 1988 paper on "Mathematical Logic in Artificial Intelligence," but the intuition is very old. A representation is *elaboration tolerant* to the extent that it is easy to change in order to reflect new information. Natural language is the ultimate model for elaboration tolerance; in everyday discourse one can easily and succinctly change one's declarations by adding an extra sentence. Elaboration tolerance is a desirable property of any knowledge representation, whether it is a logical theory, a logic-based knowledge base like Cyc, a computer program, a frame system, a Bayesian network, a neural network, or any other AI formalism. But this desirable property extends beyond AI representations – any system which operates on facts asserted declaratively will be easier to update, and to use and understand, if it is elaboration tolerant. Hence this concept applies to software systems and the World Wide Web as well.[1]

Abstractly, we can view elaborations as operations e which map a particular representation R to some desired representation R': [2]

[1] If we are successful, our theory of elaboration tolerance will follow in the footsteps of the best ideas to which AI has given birth, and be adopted by mainstream computer science. Examples include the theory of computation, timesharing, LISP, ...

[2] Belief revision and reformulation are two fields that use this same method of studying representations, albeit for different purposes. We provide an analysis of how these fields differ from our study of elaboration tolerance in Chapter 10.

$$e \cdot R \quad = \quad R' \tag{1.1}$$

The more natural and simple our operation e can be to map R to R', the more elaboration tolerant R is. Note that e could be any kind of operation, such as adding extra facts to R, deleting facts of R, or even rewriting parts of R. The more complicated e is, the more akin it is to performing "brain surgery" on R. This neurosurgery is undesirable because it requires the elaborator to understand more details of R. This defeats the spirit of elaboration tolerance, where changes are easily implemented.

1.1 The Necessity of Elaboration Tolerance

A declarative approach to AI, where all knowledge is explicitly represented by sentences and "thinking" involves inferences made on these sentences, will require elaboration tolerance in order to be useful. [McCarthy, 1959] describes the *Advice Taker*, a declarative formalism that operates solely by additions of sentences. These sentences advise how to further manipulate other sentences. [McCarthy, 1959] gives some reasons why such a system would be a good architecture for artificial intelligence (directly quoted):

1. If one wants a machine to be able to discover an abstraction, it seems most likely that the machine must be able to represent this abstraction in some relatively simple way.

2. Declarative sentences have logical consequences and it can be arranged that the machine will have available sufficiently simple logical consequences of what it is told and what it previously knew.

3. The meaning of declaratives is much less dependent on their order than is the case with imperatives. This makes it easier to have after-thoughts.

4. The effect of a declarative is less dependent on the previous state of the system so that less knowledge of this state is required on the part of the instructor.

The declarative AI approach has proven to be a viable research program. Cyc [Cyc, 2003] is a huge knowledge base containing axioms representing common sense knowledge. It has been successfully utilized in industry, as well as in recent DARPA research programs such as HPKB [Cohen et al., 1998]. The Semantic Web [Semantic Web, 2002] is a very recent undertaking to represent information on the internet so that it can be used by machines. On the other hand, without much formal annotation, search engines such as Google [Brin and Page, 2003] have been able to extract meanings from billions of web pages. For example, the text query "flowers" returns documents about flowers, including where to buy them, how to grow them, etc.

But any declarative formalism will have to be constantly updated. It is naive for us to believe we can build a non-trivial representation that is perfectly omniscient. Quite the contrary, any such representation will always be only partially complete and correct, for many reasons:

For one, language is inherently vague, often requiring re-iteration of utterances. This concept is captured by *approximate objects and theories* [McCarthy, 2000]. An object is approximate in at least two ways. The first is when details about a concept are purposefully left out, such as a summarization. The second is more subtle, in that the concept is inherently incomplete, and there is no way to ascertain the truth of certain facts about that concept. Examples of these include the number of grains that determine a heap, or the welfare of a chicken [McCarthy, 2002a].

Approximate entities abbreviate facts, so that one is not distracted by irrelevancies or inadequacies, thereby enabling efficient reasoning. But when necessary, they can be mapped to more or less approximate concepts. This movement from an approximate entity to a less approximate entity (or vice versa) is essentially an elaboration, since we are adding more details or refining a concept. We suspect that approximate objects can only be represented by means of elaboration tolerant formalisms, as they are inherently elaboratable. Elaboration tolerant representations will provide a platform from which we can better understand the formal nature of approximation.

There is a related need for easy update. [Kent, 1978] notes that there are (at least) a denumerable number of concepts, but we only have a finite number of words to describe them. A word will correspond to "a cluster of more or less related concepts." Thus, there will inevitably be confusion between a speaker's and listener's intended meaning of an utterance. In this case, elaborations will be absolutely necessary to convince the listener of the intended meaning.

Finally, as science progresses, new facts will be added to man's repository of knowledge. Intelligent systems must be able to absorb these new facts seamlessly to be useful.

These factors explain why any human-level AI declarative representation will require elaboration tolerance. Touching on some of these arguments, [McDermott, 1978] hints at the elegant solution of additive elaboration tolerance:

> After all, a practical program will never be "finished"; it would be nice to know that whatever fragment of one exists will maintain its integrity as new rules are added or new applications are made of old rules. We would like our programs to be "additive," that is, to be able to assimilate new, correct rules from experts without destroying the correctness of old ones.

1.2 Additive Elaboration Tolerance

Additive elaboration tolerance is concerned with representations that admit only one modality for change: adding formulas. A representation has *additive elaboration tolerance* if natural changes can be applied to it, *simply by adding the appropriate formula*. In the language of (1.1), R has additive elaboration tolerance when we can map R to R' by the operation: [3]

$$e \cdot R \;\; = \;\; \psi_e(R) \cup R = R' \tag{1.2}$$

This simple means of altering representations is desirable for many reasons. For one, it is the means by which human discourse works; certainly a sentence, once uttered, cannot ever truly be retracted. When speaking, one can only add statements

[3] ψ_e depends on R in (1.2) because ψ_e could be context-sensitive – its construction could depend on the structure of R.

to change the interpretation of previously uttered sentences. The second reason stems from the previously mentioned *Advice Taker* formalism:

> The *advice taker* is a proposed program for solving problems by manipulating sentences in formal languages. The main difference between it and other programs or proposed programs for manipulating formal languages (the *Logic Theory Machine* of Newell, Simon and Shaw and the Geometry Program of Gelernter) is that in the previous programs the formal system was the subject matter but the heuristics were all embodied in the program. In this program the procedures will be described as much as possible in the language itself and, in particular, the heuristics are all so described.

We want our elaborations to be expressible within the language of the target representation, so that we can reason about them as well. Additive elaboration tolerance forces this to be the case, as in this paradigm, all elaborations must be *declaratively specified, within the logic.* Ultimately, we are pushing all of our operations on R one level down into the language itself. Thus the problem of additive elaboration tolerance reduces to studying the expressivity of a representation and how it reacts to the addition of formulas. [4]

But why is this such a worthy goal? We continue our perusal of [McCarthy, 1959]:

> The main advantages we expect the advice taker to have is that its behavior will be improvable merely by making statements to it, telling it about its symbolic environment and what is wanted from it. *To make these statements will require little if any knowledge of the program or the previous knowledge of the advice taker.* [Our emphasis.]

By formulating our elaborations within the language, we absolve the user of having to perform "brain surgery" on the representation in order to bring about some elaboration. Instead of tinkering with the formulas of a given representation, and

[4] In this way, the meta-level semantics become part of the representation.

then ensuring that the result has the intended interpretation, all he must do is add the formula corresponding to the meaning of the elaboration to the set of axioms. It is the semantics' job to make sure that the formula properly encodes the change.[5] This declarative mechanism, of representing our elaborations by adding formulas, thus provides the ultimate *abstraction barrier* for altering representations.

From a practical point of view, this approach is also the best to follow. Any elaborations more complicated than additions of formulas would strongly depend upon the structure of the formalism itself. We would be forced to study how a particular syntactic transformation alters the semantics of a representation. In general, this will require intricate understanding equivalent to that required for brain surgery. We explore these issues more formally in Chapter 3.

1.3 Thesis Roadmap

We have divided this thesis into three parts. Part I introduces and motivates the problem of elaboration tolerance in general. Some previous investigations of elaboration tolerance are explored in Chapter 2. Chapter 3 introduces a formal framework for talking about elaboration tolerance, and shows how certain design principles can work to promote this property.

Part II focuses on additive elaboration tolerance. Chapter 4 defines additive elaboration tolerance in terms of two fundamental classes of elaborations: full retraction and full addition. Full retraction is the ability to retract the truth of a formula, by adding a formula. Full addition adds a formula without introducing inconsistency, and can be retractable later. We extend the framework of Chapter 3 to this paradigm, using the distinction between hard (absolutely true) and soft (probably true) facts. Elaborations are defined as those operations which change the soft facts, and preserve the hard ones.

The next chapters formally show how to endow arbitrary systems with additive elaboration tolerance by embedding it in a more powerful system. Section 4.4 provides a friendly introduction, sketching out the general principles involved without mucking

[5]We can describe our approach by the maxim "Knowledge base, know thyself."

with the technical details. That is the task of the remaining chapters of Part II:

Chapter 5 introduces the mathematical infrastructure we use to generally characterize our representations. We call these structures *extended axiomatic formal systems with choice*. Chapter 6 shows how one can combine two such representations S_1 and S_2 to produce another system $S_{1,2}$. We also provide a semantics for $S_{1,2}$, based on the cross-product of models. The motivation is that this way of combining semantics is itself an elaboration tolerant means to add elaboration tolerant features to a formalism. Chapter 7 introduces a special scaffolding S_{Ab} that is used to augment a representation S_1 to a new one $S_{1,Ab} = S_{Ab} \cdot S_1$ that has additive elaboration tolerance. Chapter 8 formally proves this, by showing $S_{1,Ab}$ has full retraction and full addition for any formula in the original language $\mathcal{L}_1$ of S_1.

$S_{1,Ab}$ was originally only meant to be a simple witness that an arbitrary representation can be augmented to have additive elaboration tolerance. Chapter 9 explores some serendipitous properties of the construction. The chapter also provides some reassuring results, such as that $S_{1,Ab}$ is a sound construction, and that it initially retains (before any elaborations) the same hard and soft formulas that were derivable from S_1. We also discuss some other desirable features, such as the fact that $S_{1,Ab}$ is expressive enough to encode and respect the *relevance* of formulas. It turns out that we can also "compactify" our representation after a sequence of elaborations are applied. Let $\psi_n \cup \ldots \cup \psi_1 \cup S_{1,Ab}$ denote the system after n elaborations. By compaction we mean we can find a representation $S'_{1,Ab}$ with fewer axioms which is equivalent, even under subsequent elaborations.

Finally, Part III examines the ramifications of elaboration tolerance including its relevance to other fields, including belief revision. The famous AGM postulates can be adapted to our own $S_{1,Ab}$, resulting in some intuitive restrictions. Chapter 10 also relates elaboration tolerance to reformulation, Bayesian networks, and neural networks.

Chapter 11 muses about the philosophical aspects of elaboration tolerance, and argues that it is absolutely necessary for the further progress of the logical AI program. We also recount some interesting relationships between elaboration tolerance and the work done on the frame problem. In Section 11.4, we explore a weakness of our system

$S_{1,Ab}$: its elaborations are limited in their resolution in the sense that they cannot be retracted or added on a case-by-case basis. [McCarthy, 2003] points out that his system formalizing which birds fly in [McCarthy, 1986] does have this property. In other words, a rule can be retracted for a particular instance, but still be made to hold in place generally. We explore this example further, as well as the more general concept of viewing nonmonotonic reasoning as an agent of elaboration tolerance. We conclude with some interesting and practical research directions in Chapter 12.

We have put the bulk of the proofs in Appendix A to make this work easier to absorb. To make the thesis more accessible, we have included an index as well.

Chapter 2

Previous Work on Elaboration Tolerance

2.1 Introduction

Elaboration tolerance is first noted in John McCarthy's 1988 paper [McCarthy, 1988], and there are glimmers of the idea in the seminal 1959 paper, "Programs with Common Sense" [McCarthy, 1959]. However, it is not until almost a decade later in [McCarthy, 1997] that elaboration tolerance is explicitly studied on its own. This paper further provides a typology of elaborations, and introduces the missionaries and cannibals problem with its many elaborations as a *Drosophila* for the subject. In that same year, Murray Shanahan in his book *Solving the Frame Problem* relates the importance of elaboration tolerance to the *frame problem*. A year later, Eyal Amir provides a syntactic framework within which to study and quantify elaboration tolerance, described in [Amir, 1998].

We also find two approaches to designing elaboration tolerant representations in the realm of logic programming. [Gelfond and Przymusinska, 1996] examine elaboration tolerance in terms of changing the input/output specifications of programs, while [Alferes et al., 2000] demonstrate a method for updating a logic program with another one. There are also two rather successful attempts at finding elaboration tolerant representations of McCarthy's missionaries and cannibals problem: [Lifschitz, 2000] and [Gustafsson and Kvarnström, 2001].

2.2 The Original Paper on Elaboration Tolerance

[McCarthy, 1997] introduces elaboration tolerance and explains its necessity for human-level AI: intelligent systems will need "to be able to take new phenomena into account," without having to rebuild their knowledge from scratch. McCarthy also identifies different kinds of elaborations, citing addition of formulas as the simplest kind, but also considering changing values of parameters, changing the arity of relations and functions, and so on.

[McCarthy, 1997] makes two key observations. The first is that elaboration tolerance will require nonmonotonic reasoning (at least for additive elaboration tolerance). The second is that a meta-level representation of the facts will be needed in order to effect elaboration tolerance.

Another contribution of the work is the introduction of the missionaries and cannibals problem (MCP) as a *Drosophila* for the subject:

> Three missionaries and three cannibals come to a river and find a boat that holds two. If the cannibals ever outnumber the missionaries on either bank, the missionaries will be eaten.
>
> How shall they cross?

From this description, one can create a formalization in which to compute the movements of the missionaries and cannibals so that they all cross safely. [McCarthy, 1997] then submits nineteen[1] different elaborations of the problem, such as changing the number of missionaries and cannibals involved, or adding the fact that a bridge fords the river. These elaborations serve to benchmark any prospective formalization of the MCP: it should be easy to alter the representation (ultimately by only adding sentences) to reflect the change and still solve the problem of crossing the river.

[1] As of April 1, 2003, there are now twenty.

McCarthy also provides a typology of elaborations. Some examples include adding preconditions to actions, making properties depend on a situation, going into detail, and splitting an entity.

2.3 Elaboration Tolerance and the Frame Problem

Murray Shanahan in his book *Solving the Frame Problem* [Shanahan, 1997] provides a grand overview of many different representations of action and change, and how they work to solve the *frame problem*. The frame problem is that of concisely representing not only what facts change after an action occurs, but which ones remain the same. Shanahan gives three criteria for a satisfactory solution to the frame problem:

1. representational parsimony
2. expressive flexibility
3. elaboration tolerance

Representational parsimony means that the representation of the effects of actions should be compact, with size on order of the complexity of the domain. Expressive flexibility refers to the ability to represent arbitrary phenomena in the framework, such as concurrent actions or continuous change. Shanahan defines elaboration tolerance as when

> the effort required to add new information to the representation is proportional to the complexity of that information. In particular, to augment a situation calculus theory with a new action that directly affects, say, n fluents, might require the addition of roughly n new sentences, but it should not necessitate the complete reconstruction of the old theory. Rather, facts about the effects of the new action should be gracefully absorbed into the old theory.

In an elaboration tolerant representation, the complexity of the changes only depends on the information to be changed. Shanahan goes on to echo [McCarthy, 1997]'s preference for an additive modality:

> Ideally elaboration tolerance would mean that a new sentence could be appended directly to the old theory to yield the new one. Appending a single sentence to a theory is the simplest possible modification of that theory.

This observation reflects our own fascination with additive elaboration tolerance. In fact, almost every approach to the frame problem has shared our attraction, focusing on solutions which only add formulas which prescribe the new effect of an action to the set of axioms. This is evident in causal approaches such as [Haugh, 1987] and [Lifschitz, 1987] and state-based ones such as [Baker, 1991]. This modality also shows up in the action languages [Gelfond and Lifschitz, 1993], [Giunchiglia and Lifschitz, 1998], and [Kakas et al., 1999], as well as the event calculus formalism of [Shanahan, 1997]. The only solutions which do not use addition of formulas along with some requisite nonmonotonic reasoning are the explanation closure axioms of [Haas, 1987] and [Schubert, 1990], and the successor state axioms of [Reiter, 1991] and [McIlraith, 2000]. After an examination of the literature on reasoning about actions and change, it appears that of the three above criteria, elaboration tolerance (particularly with an additive mode) has been the most important factor in constructing solutions to the frame (and ramification and qualification) problem.

Shanahan also observes that elaboration tolerance requires nonmonotonicity. He notes that most solutions to the frame problem infer the persistence of a fluent if it is not affected by an action. But the solution should also tolerate the addition of new effects of actions, and thus should inherently acknowledge that it does not represent *all* known effects of all actions. Additive elaboration tolerance will require any accompanying consequence relation to be nonmonotonic, as some facts may not be true after the addition of a new axiom.

2.4 A Syntactic Approach to Elaboration Tolerance

Eyal Amir in [Amir, 2000] investigates elaboration tolerance in terms of the syntactic structure of theories. He uses *axiomatic formal systems* of the form $\langle \mathcal{L}, \mid\!\sim, \Gamma \rangle$ to represent knowledge bases.[2] Elaborations are restricted to sequences of the addition and deletion of formulas in the language. Since knowledge bases may be expressed in different languages, Amir uses a translation function t to translate two axiomatic formal systems to a "common ground." Arbitrary formalisms can then be compared via their translated versions under t. One representation is (syntactically) more elaboration tolerant than another if it takes fewer actions to change it to some target representation mapped to by t.

[Amir, 2000] also presents the notion of the *abnormalization* of a theory, where each formula ϕ in a theory is replaced by the disjunction $Ab_\phi \vee \phi$. Then logical consequence under circumscription, where the Abs are minimized, is taken as the consequence relation. It turns out that this paradigm requires fewer additions of axioms than the original theory in order to reach a target representation.

Amir goes on to show that a propositional formalism is less elaboration tolerant than another which has more (by superset inclusion) symbols, when elaborations are restricted to only adding formulas. Also, under this same restriction of elaborations, a formalism is less elaboration tolerant than its abnormalized version. However [Amir, 2000] demonstrates that there are other kinds of nonmonotonic theories that are less elaboration tolerant than their monotonic equivalents. Finally, given one axiomatic formal system, one can always find another that is strictly more elaboration tolerant (under Amir's criterion) for a certain target representation.

[2] Axiomatic formal systems are comprised of a language $\mathcal{L}$, set of axioms Γ which are in this language, and a consequence relation $\mid\!\sim$ which operates on sets of $\mathcal{L}$-formulas.

2.5 Approaches Based on Logic Programming

2.5.1 Specifications of Programs

[Gelfond and Przymusinska, 1996] describe software development as the process of moving from a specification, to a representation, to the implementation. The research focuses on the transition from specification to representation. Gelfond and Przymusinska formally define a *specification* as a description which shows how to map literals in one language $\mathcal{L}_{source}$ to literals in another language $\mathcal{L}_{target}$. The *representation* of the specification is modeled by declarative logic programs using stable model semantics. The representation is meant to realize the specification by applying the logic program to literals from $\mathcal{L}_{source}$. The resulting inferences, restricted to the language $\mathcal{L}_{target}$, should match what was intended by the specification. An *lp-function* refers to the combination of the logic program and $\mathcal{L}_{source}$-literals.

Gelfond and Przymusinska assert that a representation of a specification is elaboration tolerant if a small change to the specification requires a small change to the accompanying representation. They believe that in practice, specifications will be built up functionally from simpler ones, and that their representations will be *homomorphic*. This means that if we create a new specification $spec_{new} = f(spec_1, spec_2)$, and Π_1 is the representation for $spec_1$, and Π_2 that for $spec_2$, then the representation for $spec_{new}$ will be some function g of Π_1 and Π_2. The elaboration tolerance of $spec_1$ and $spec_2$ with respect to f is measured in terms of the complexity of this g.

The paper introduces two homomorphic kinds of combinations: *incremental extensions* and *simple input extensions*. Incremental extensions are simply the union of two specifications, where neither of the source and target languages interact adversely. In this case, the representation of incremental extensions is just the union of the logic programs. Simple input extensions are specifications which are extended to accept as its input formulas both in $\mathcal{L}_{source}$ as well and $\mathcal{L}_{target}$. The representation for this kind of extension is created by separating the logic program into two parts, altering one of them, and then recombining them.

2.5.2 Dynamic Logic Programming

[Alferes et al., 2000] introduce the paradigm of *dynamic logic programming*, which shows how to update a logic program P by another program U, resulting in the program $P \oplus U$. This paradigm is used to deal with knowledge that evolves with time. Their treatment not only allows one to change facts (the *extensional* part of the logic program), but the rules (the *intensional* part).

Alferes et al. use *propositional Horn theories* as their representational substrate, where there is at most one atom or negated atom in the head of each rule. This means that every rule in a logic program P is either of the form:

$$\begin{array}{rl} \texttt{not a} & \texttt{:- c}_1\texttt{, ..., c}_\texttt{m}\texttt{, not d}_1\texttt{, ..., not d}_\texttt{n} \\ & \textit{or} \\ \texttt{a} & \texttt{:- c}_1\texttt{, ..., c}_\texttt{m}\texttt{, not d}_1\texttt{, ..., not d}_\texttt{n}\texttt{,} \end{array} \tag{2.1}$$

where each $\texttt{a}$, $\texttt{c}_\texttt{i}$, and $\texttt{d}_\texttt{j}$ is a member of a set of proposition letters $\mathcal{P}$. *Stable model semantics* are used to give meaning to these programs, where `not` is the default negation operator.

Given two propositional Horn logic programs P (original program) and U (update), the *update* $P \oplus U$ consists of the following sentences in the expanded language $\mathcal{L}^+ = \mathcal{P} \cup \{\texttt{a}^-,\ \texttt{a}_\texttt{P},\ \texttt{a}_\texttt{P}^-,\ \texttt{a}_\texttt{U},\ \texttt{a}_\texttt{U}^- \mid \texttt{a} \in \mathcal{P}\}$:

1. (RP) Rewritten original program clauses:

   (a) For each $\texttt{a :- c}_1\texttt{, ..., c}_\texttt{m}\texttt{, not d}_1\texttt{, ...not d}_\texttt{n} \in P$, we have $\texttt{a}_\texttt{P}\texttt{ :- c}_1\texttt{, ..., c}_\texttt{m}\texttt{, d}_1^-\texttt{, ...d}_\texttt{n}^- \in P \oplus U$.

   (b) For each $\texttt{not a :- c}_1\texttt{, ..., c}_\texttt{m}\texttt{, not d}_1\texttt{, ...not d}_\texttt{n} \in P$, we have $\texttt{a}_\texttt{P}^-\texttt{ :- c}_1\texttt{, ..., c}_\texttt{m}\texttt{, d}_1^-\texttt{, ...d}_\texttt{n}^- \in P \oplus U$.

2. (RU) Rewritten updating program clauses:

   (a) For each $\texttt{a :- c}_1\texttt{, ..., c}_\texttt{m}\texttt{, not d}_1\texttt{, ...not d}_\texttt{n} \in U$, we have $\texttt{a}_\texttt{U}\texttt{ :- c}_1\texttt{, ..., c}_\texttt{m}\texttt{, d}_1^-\texttt{, ...d}_\texttt{n}^- \in P \oplus U$.

   (b) And for each $\texttt{not a :- c}_1\texttt{, ..., c}_\texttt{m}\texttt{, not d}_1\texttt{, ...not d}_\texttt{n} \in U$, we have $\texttt{a}_\texttt{U}^- \texttt{ :- c}_1\texttt{, ..., c}_\texttt{m}\texttt{, d}_1^-\texttt{, ...d}_\texttt{n}^- \in P \oplus U$.

3. (UR) Update rules:

   For each $\texttt{a} \in \mathcal{P}$,

   (a) $\texttt{a :- a}_\texttt{U} \in P \oplus U$.

   (b) $\texttt{a}^- \texttt{ :- a}_\texttt{U}^- \in P \oplus U$.

4. (IR) Inheritance rules:

   For each $\texttt{a} \in \mathcal{P}$,

   (a) $\texttt{a :- a}_\texttt{P}\texttt{, not a}_\texttt{U}^- \in P \oplus U$.

   (b) $\texttt{a}^- \texttt{ :- a}_\texttt{P}^-\texttt{, not a}_\texttt{U} \in P \oplus U$.

5. (DR) Default rules:

   For each $\texttt{a} \in \mathcal{P}$,

   (a) $\texttt{a}^-\texttt{:- not a}_\texttt{P}\texttt{, not a}_\texttt{U}$

   (b) $\texttt{not a:- a}^-$

The sets of rules (RP) and (RU) rewrite the heads of all rules in terms of $\texttt{a}_\texttt{P}$, $\texttt{a}_\texttt{P}^-$, $\texttt{a}_\texttt{U}$ and $\texttt{a}_\texttt{U}^-$. This allows us to distinguish from which program, P or U, each conclusion hails. With this distinction in place, one can prioritize the conclusions by means of rules in (UR) and (IR) – if a conclusion follows from U, then accept it immediately; if P espouses some conclusion accept it only if U does not contradict it. Finally, the rules (DR) translate the negative atoms $\texttt{a}_\texttt{P}^-$ and $\texttt{a}_\texttt{U}^-$ back to their negated counterparts – in any particular model, $\texttt{a}^-$ holds as long as neither $\texttt{a}_\texttt{P}$ nor $\texttt{a}_\texttt{U}$ are inferred, which means that `not a` will have to hold as well.

Alferes et al. note that the construction of $P \oplus U$ from P and U takes at most linear time in the size of the programs. Theorem 4.1 shows how any stable model of $P \oplus U$ can be constructed very straightforwardly from rules of $P \cup U$. [Alferes et al., 2000] go on to generalize their notion of program update to a sequence or even tree of programs, each of which updates their predecessor.

The idea of rewriting atoms in a given theory and applying some kind of non-monotonic reasoning to update it is not new. [Winslett, 1989] performs a similar transformation on a logical theory. The theory consists of protected and unprotected formulas; the protected formulas are those which are never to be retracted and must hold true in the updated theory. The theory T in question is transformed to another theory, where each occurrence of the predicate symbol P is replaced with the symbol $oldP$, and a copy of only the protected formulas is added with P replaced by $newP$. To update the theory with α, the formula α' is added, which is α with every P replaced with $newP$. Then, the theory is circumscribed, minimizing the change between $newP$ and $oldP$. [3]

As an aside, the approach of [Alferes et al., 2000] is semantically different from that of [Winslett, 1989], in how they update their formalisms. Winslett's approach follows the *interpretation update* model, proposed in [Katsuno and Mendelzon, 1992]. In this paradigm, we first take the models of the original database DB, update each of them individually, and then define DB' as the database which corresponds to the set of updated models. [Alferes et al., 2000] notes that interpretation update is computationally impractical, and can lead to unintuitive results, particularly when an update is meant to alter the behavior of the intensional part of the original database. This method also does not respect the *syntax* of a program, which could encode important information (particularly causal links). [Alferes et al., 2000]'s approach is actually closer to that of *revision*, where the new database DB' corresponds to those models of the update which are closest to the original database DB. This approach is the same as interpretation update when P is purely extensional. We discuss this topic of update versus revision in more detail in Section 10.1.1, and how it relates to elaboration tolerance.

[3] It is slightly more complicated than this. Winslett also gives each predicate symbol P a priority, and *prioritized circumscription* is performed over the theory, respecting these priorities.

2.6 Elaboration Tolerant Solutions to the MCP

2.6.1 The Causal Calculator

CCALC [McCain and Turner, 1997] is a program used for reasoning about actions and change. It uses the principle of causality to determine the relations between fluents and actions, which respectively hold/occur at integral timepoints. It has effect axioms of the form:

$$a \texttt{ causes } f \texttt{ if } \phi, \tag{2.2}$$

which means that if action a occurs at time t and formula ϕ holds at t, fluent f will told at $t+1$. Ramifications are handled by the directive:

$$\texttt{causedby } f \texttt{ if } \phi, \tag{2.3}$$

which states that f holds at t if ϕ holds at t. Finally, "anti"-preconditions are modeled by the statement:

$$\texttt{nonexecutable } a \texttt{ if } \phi, \tag{2.4}$$

asserting that if ϕ holds at t then a cannot occur at t. Unlike most action formalisms, CCALC assumes an action is executable as long as one cannot prove it otherwise. These three statements comprise what are known as *action descriptions*. Note that a, f and ϕ can be parameterized by free variables.

CCALC is really only a *high-level description language*. CCALC files are compiled by a Prolog program to a grounded version, which implements the nonmonotonic causal mechanisms of the language. CCALC also includes other helpful constructions:

First of all, the action descriptions are listed in files, and CCALC allows for directives which can load in statements from another file into the current one. This mechanism thus treats each file as a *context*, with the file loading directive corresponding to a *lifting axiom*.

Sorts are the second important feature: each term, including variables, must be declared ahead of time, and assigned a particular sort (or type). Then during compilation, variable instantiation is restricted only to constants of that sort, giving rise to a version of the closed world assumption. Every new declared object automatically inherits the relevant properties of the sort. Some pre-determined sorts have special [nonmonotonic] semantics and are used to solve the frame problem. For example, fluents of type `inertialTrueFluent` are meant to remain true after an action, if no other information is available.

Third, *labels* are added to identify defeasible statements; these can be used to nullify statements later on. CCALC statements of the form (2.5):

$$(\texttt{label1})\ \Phi, \tag{2.5}$$

are compiled to rules of the form (2.6):

$$\neg Ab(\texttt{label1}) \implies \Phi. \tag{2.6}$$

The semantics of the Prolog implementation assumes $\neg Ab(\ell)$ for any label ℓ by default, so unless $Ab(\texttt{label1})$ is asserted, Φ will hold in the program.

These three features of CCALC promote elaboration tolerance. Having file loading directives allows one to cleanly organize action descriptions. The sorts nicely restrict grounding only to pertinent terms. The labels allow one to have *defeasible* action descriptions.

[Lifschitz, 2000] uses CCALC to represent McCarthy's elaborations of the missionaries and cannibals problem. In addition to the elaboration tolerant features of CCALC, Lifschitz also uses syntax and choice of language to his advantage. For example, one important kind of elaboration is adding an attribute to a term, such as the speed at which the action $cross(bank1, bank2)$ is achieved. Instead of surgically inserting another argument to say something like $cross(bank1, bank2, quickly)$, one adds the fact as an attribute, or fluent:[4]

[4]The syntax used in [Lifschitz, 2000] is slightly different, but this is the general idea. This concept is also known as *operator splitting*.

$$cross(bank1, bank2) \texttt{ causes } speed(cross(bank1, bank2), quickly) \texttt{ if } \top.$$

This *action attribute* uses an analytic rather than synthetic syntax [McCarthy, 1962] to elaborate an action. One can see that the analytic syntax is much more elaboration tolerant, as it only requires an expansion of the language to express another attribute of an object, as opposed to the rewriting of a symbol.

A second important aspect of Lifschitz's formalization is the use of groups, or sets, of objects. Lifschitz uses the fluent $num(set, location, number)$ to assert that *number* of the elements in *set* are at *location*. This construct allows easy representation of any new constructs, such as the set of missionaries who can row, the largest cannibal, etc.

In order to address ten out of the original nineteen elaborations of the missionaries and cannibals problem, Lifschitz first creates a file describing the basic missionaries and cannibals problem. Each elaboration consists of a new file which loads in the basic problem, adds the extra constraints associated with the elaborations (mainly with the use of `nonexecutable`), and retracts (using the labels) any statements inconsistent with the new elaboration. Action attributes are used to elaborate the actions, and constraints added over their use. Concurrency is a part of CCALC so adding *non-interacting* concurrent actions is straightforward.[5]

2.6.2 Object-Oriented Temporal Action Language

[Gustafsson and Kvarnström, 2001] introduce an intriguing object oriented framework in which to solve the missionaries and cannibals problem. It operates using a variant of a TAL (Temporal Action Logic) [Doherty et al., 1998], which itself is based on two components: $\mathcal{L}(ND)$, a high level description language, which compiles down

[5]The problem of "difficult concurrency," where two concurrent actions effect the same fluent cumulatively, has been addressed by *additive fluents* [Lee and Lifschitz, 2001]. Additive fluents can correctly represent the number of people moved across when there are multiple, concurrent moves, which cumulatively affect the same fluent. Examples include the missionaries and cannibals problem when there is a bridge, or when Jesus walks on water alongside the boat.

to $\mathcal{L}(FL)$, a low level language for implementing the actual inferences.

The framework consists of *classes*, *objects*, and *attributes*. Classes inherit properties from other classes, and have certain attributes (fluents). Objects are instantiations of classes. *Methods* are executions of code in the usual object-oriented paradigm, but here they correspond to TAL formulas that must be true at invocation time. Methods can be used to set values of fluents, constrain them to certain values, and access their values.

There are three ways [Gustafsson and Kvarnström, 2001] use these mechanisms to elaborate a particular domain. First, more detailed subclasses can be introduced which inherit properties from parent classes. Second, attributes and methods can be added to a class. Thirdly, these methods can be further overridden, with the help of a special fluent `override` that is used like *Ab* to inject defeasibility into theories: each method of the form $a.b = c$ is compiled to a statement of the form

$$\neg\texttt{override}(a, b, c) \implies a.b = c,$$

where `override` is a special predicate that by default is assumed to be false.

To their credit, Gustafsson and Kvarnström model the original missionaries and cannibals problem first, without peeking at the nineteen elaborations. Their object-oriented paradigm formalizes fourteen of the nineteen elaborations, and can handle concurrent actions which affect the same fluent. Their primary means of elaboration is through the creation of overriding subclasses, and then adding objects to these more specific subclasses. The other way is by adding attributes, and additional constraints on them through methods.

2.6.3 Evaluation of the Two Approaches

[Lifschitz, 2000] and [Gustafsson and Kvarnström, 2001] share some common features that enable elaboration tolerance:

1. *Defeasibility*: Adding facts is easy; one needs to be able to retract them as well. Both use the same mechanism to accomplish this. Lifschitz uses abnormalities,

while Gustafsson and Kvarnström use the fluent `override`, along with some built-in nonmonotonicity.

2. *Contexts*: some context-like mechanisms are used to consolidate and organize axioms. While Lifschitz uses different files, Gustafsson and Kvarnström use classes/subclasses. Both employ an inheritance-like mechanism to combine facts. Contexts are powerful because they provide a handle to refer to a collection of formulas.

3. *Sorts*: In both formalisms they not only act to enforce the closed-world assumption, but organize information to some extent. They can be used to restrict variable substitutions within the language.

4. *Attributes*: Features are easily added to terms by simply attributing them, using extra fluents. The object oriented model admits easy addition of attributes to classes, so that all objects inherit the attributes.

5. *High level versus low level implementation*: A key feature of both formalisms is that they employ a high level language to describe what is desired without cluttering up the formalism with the nitty-gritty details of implementation. That is the task of the low-level language, which actually computes the solution.

Chapter 3 will introduce a formal framework within which we can explain how the above constructs promote elaboration tolerance.

Chapter 3

General Principles of Elaboration Tolerance

In this chapter, we provide a natural framework for representations and their elaborations, and use it to find general principles of elaboration tolerance. Described in Section 3.1, this framework is based on the distinction between the hard and soft consequences of a particular representation R. Elaborations are defined as only changing the soft conclusions of a representation. Section 3.2 then asserts two requirements for elaboration tolerance: easy alteration and easy prediction. It is not enough that our elaborations be simple to implement; the consequences of the change should be somewhat obvious. We focus on how these two principles affect the formal properties of our framework in Section 3.3. This study leads to various design principles that can promote elaboration tolerance, described in Section 3.4.

3.1 Initial Framework

Consider [McCarthy, 1997]'s missionaries and cannibals problem:

> Three missionaries and three cannibals come to a river and find a boat that holds two. If the cannibals ever outnumber the missionaries on either bank, the missionaries will be eaten. (3.1)
>
> How shall they cross?

This description (3.1) has some associated representation R. Furthermore, R can be used to solve the puzzle, showing how the actors can cross the river in eleven moves, without anyone being eaten. But then, one can further elaborate the puzzle by appending $\psi_{3\to4}(R)$:

> Assume instead that there are four missionaries and four cannibals. (3.2)

Let us call this updated description $\psi_{3\to4}(R) \cdot R$. Any person will know that this new problem has the same meaning as:

> **Four** missionaries and **four** cannibals come to a river and find a boat that holds two. If the cannibals ever outnumber the missionaries on either bank, the missionaries will be eaten. (3.3)
>
> How shall they cross?

With the addition of one simple statement, the entire formulation has changed to include a different number of actors. Furthermore, humans can readily use this updated representation $\psi_{3\to4}(R) \cdot R$ to solve the altered missionaries and cannibals problem with four actors.

The point of this example is that we have changed the intended meaning of our representation, in this case, by simply adding a statement. This elaboration is succinct, and the result contains enough information to solve the altered problem. This ease of change is in fact part of the appeal of natural language. Our question is, how do we design our formal representations so that they also admit this easy elaboration tolerance? Answering this question is the main thrust of this thesis.

We first observe that we do not know how R is formally represented. Whatever R is, it does embody the notion in (3.1), and conversely the words in (3.1) are enough to specify R. But, it is not likely that R consists exactly of these English statements. [McCarthy, 1997] gives the reason why: a person, if asked to repeat the problem, is unlikely to give the same sequence of words as in (3.1). R has some internal representation in our brains that we cannot directly access. For now, we will remain agnostic about what R looks like.

But while we may not know the exact structure of R, we can *infer* all sorts of facts from it. After hearing (3.1), we know that there is a river that needs to be crossed, without anyone being eaten. Also, there is probably not a bridge. This second inference is a *Gricean implicature* [Grice, 1989]. This is an inference based on a rule of conversation where if a fact that is relevant is not mentioned, it must not hold. Formally, we can represent these inferences using the following notation:

$$\begin{aligned} &R \vdash \text{“There is a river that must be crossed without anyone being eaten.”} \\ &R \vdash \text{“There is probably not a bridge.”} \end{aligned} \tag{3.4}$$

We can use our handy consequence relation $\vdash$ as a way to represent our indirect inspection of R. But first let us examine the nature of the deductions in (3.4) a bit closer. There is a subtle difference between the nature of the first sentence derived from R in (3.4) and the second one. The first sentence is a fact which is *indisputable*. On the other hand, the fact that there is probably not a bridge is only *likely to hold*, in that it does not logically follow from R. It is more an assumption than an inference. In this case it is meta-linguistic, derived from background knowledge and the rules of communication. This second kind of inference, although weaker, is just as crucial to solving the problem as our indisputable facts.

We find it important here (and for the rest of this thesis) to formally distinguish between these two kinds of deductions. Instead of (3.4), we will write:

$$\begin{aligned} &R \vdash \text{“There is a river that must be crossed without anyone being eaten.”} \\ &R \mid\!\sim \text{“There is not a bridge.”} \end{aligned} \tag{3.5}$$

Intuitively, $\vdash$ is a stricter consequence relation than $\mid\!\sim$. $\vdash$ only produces conclusions that logically follow from R, while $\mid\!\sim$ infers facts that are only usually true. The distinction is nicely described in [Kraus et al., 1990] in terms of *hard* and *soft* truths. The facts derivable from R using $\vdash$ are the hard truths, while those using $\mid\!\sim$ are soft ones. Hard truths are more stable, in that they will remain true, no matter what new information we are given. This means that $\vdash$ is a *monotonic consequence relation*. An example of a hard truth is "1+1=2" – no matter what new information we learn, this fact will always hold.

Soft truths on the other hand are defeasible. These are facts that usually hold, but are based on defaults such as conversational rules of thumb or background information. These are often made false in the presence of additional information, such as "Tweety flies." We will use $\mid\!\sim$ to represent this kind of common sense inference, that is, the mechanism people (and smart robots) use to informally infer facts about the world day-to-day. In the case of our smart robots, $\mid\!\sim$ may not be as loosely specified. It could be something as concrete as a closed-world assumption on a database, or a communication convention in a protocol [Genesereth, 2003]. The point is that the nature of $\mid\!\sim$ depends on some sort of defeasible reasoning that is not entirely sound. However, since it gives us more information than $\vdash$ (in general we assume $\mid\!\sim\ \subseteq\ \vdash$), we use it as our primary means of inference.

There is also a useful semantic means of distinguishing between hard and soft facts. If we envision the truth of a fact ϕ as when it holds in some set of possible worlds, hard truths are those which hold in *all* worlds described by R that we can imagine, that is all states of the universe. But most of the time we do not (or cannot) consider all possible worlds, and only consider truth in the *preferred* or *most likely* subset of all possible worlds espoused by R. We will construct the formal framework that allows this semantic interpretation later in Section 5.2. For now we will assume it as given. Viewing soft truth in terms of preferred worlds is not unreasonable, as there are all sorts of principles which restrict our view of the world in ways that are not always valid. Figure 3.1 demonstrates this graphically. Formalisms such as conditional logic [Chellas, 1980] and *counterfactuals* [Lewis, 1973] [Costello and McCarthy, 1999], where truth is determined in terms of the most likely or closest worlds to the current one, are built on these same ideas.

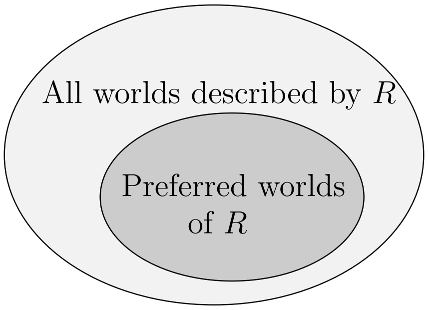

Figure 3.1: The preferred worlds of R are a subset of all of worlds described by R.

What is the importance of this distinction between hard and soft facts? We want to use this hard/soft distinction to characterize our elaborations – specifically, elaborations can only change soft facts, and must leave the hard ones as is. Intuitively, our representation R only gives us an incomplete vision of what we are trying to describe; the set of facts derived by $\vdash$ is sparse because they must be true in *all* versions of R. $\mid\!\sim$ "fills in the gaps" by providing extra facts, relying on rules of conversation and other defeasible modes of inference. We envision our elaboration e as constructing a revision of R where the gaps are filled in differently. Formally, we can write this as:

$$R \vdash \phi \implies e \cdot R \vdash \phi, \tag{3.6}$$

while it is not necessarily the case that

$$R \mid\!\sim \phi \implies e \cdot R \mid\!\sim \phi. \tag{3.7}$$

That is, while the elaboration operation $e\cdot$ preserves hard facts, it can alter soft ones.

We can extend our semantic intuition depicted in Figure 3.1 to this view of how elaborations work. Since elaborations only change the soft facts, we can view them as changing the set of preferred worlds, within the set of all worlds, as shown in Figure 3.2:

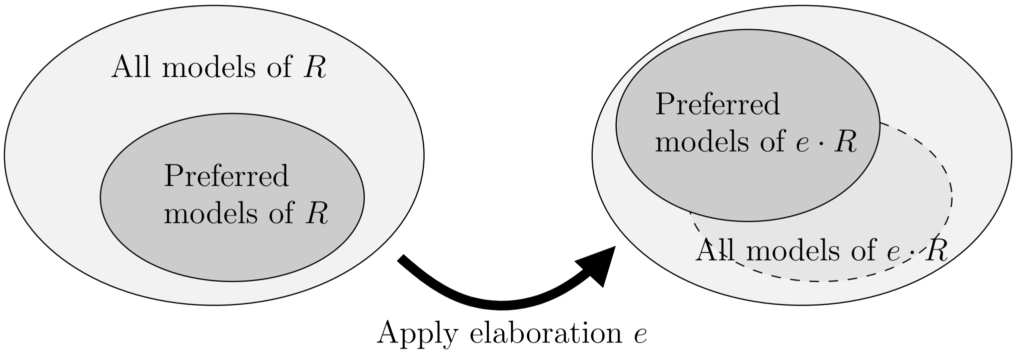

Figure 3.2: Elaborations change the focus of preferred models.

Since the hard facts espoused by R are immutable, our elaboration e must preserve their truth. Figure 3.2 verifies this by showing that the set of *all* worlds does not change after application of e (only the preferred ones do). If the set of all worlds does not change with an elaboration, all hard facts will also remain the same.

As $\vdash$ must examine each model of R to determine truth, we can use its negation to determine what facts can *possibly be true,* and thus elaborated. $\mid\!\sim$ plays the dual role of asserting that what facts are *likely.* [1] Hence, as mentioned before, we will use $\mid\!\sim$ as our primary means of generating consequences, using $\vdash$ only to find out what facts are possible.

Distinguishing between degrees of truth is a crucial part of many other different fields of artificial intelligence. [Ginsberg and Smith, 1987] uses a concept of *protected laws*, which are domain constraints that can never be violated when reasoning about the effects on an action. An example of a protected law is that an object is in exactly one place at one time. These laws play a similar role in determining what

[1] It is important to note that for a representation R in human language, every conceivable state of the universe is a possible world of R, as almost anything is possible. Thus, almost all facts are soft and thus subject to being elaborated.

facts hold and which change after an action is applied. Certainly the field of probabilistic reasoning [Pearl, 1988] is entirely concerned with quantifying the level of uncertainty of a statement. Bayes rule quantifies how the probability of a statement changes as one learns a new fact. The concept of *epistemic entrenchment* in belief revision [Gärdenfors, 1992] is used to quantify how strongly a statement is believed, which is important when revising one's beliefs. For example, say a belief ϕ is more epistemically entrenched than some other belief ψ, written $\phi > \psi$. If we ever have to choose between these two beliefs (say in a revision), we should keep ϕ and not believe ψ, as we believe it less strongly.

[Fikes, 2003c] points out that the hard/soft distinction between facts can be used to sharpen the notion of an *ontology*. An ontology is the formalization of a domain using special structures. For example, a time domain may include Allen interval relations [Allen, 1981] and calendars as fundamental entities. Often one uses special-purpose reasoners to take advantage of this built-in structure, and reason more effectively [Fikes et al., 2003]. Fikes suggests that this structure corresponds to a domain theory, which must be true in all possible worlds. This domain theory in turn can be thought of as the set of hard facts.

Later on, it might be useful to represent different kinds of facts with different consequence relations, such as $\mid\!\sim_\alpha, \mid\!\sim_\beta, \ldots$. For now, we stick to our simple paradigm of $\vdash$ and $\mid\!\sim$.

3.2 First Principles of Elaboration Tolerance

A representation R is elaboration tolerant to the extent it is easy to change to some other representation R'. This definition entails two necessary properties: that R be "easy to change," but also "easy to predict:"

1. "easy to change" – the elaborations must be of low complexity.

   By low complexity, we mean simple computations, such as only adding or deleting formulas. Or, the elaboration could be some simple syntactic manipulation of the formulas in the language. A more complicated elaboration would change the language of the representation. [2]

[Shanahan, 1997] provides a metric for elaboration tolerance, asserting that a representation is elaboration tolerant if the effort required to apply an elaboration is proportional to the complexity of information being added. More glibly, the elaboration should not require a rewrite of the entire representation.

2. "easy to predict." We must be confident that the changes we apply to our elaboration will result in the desired representation. To be precise, we use elaborations to change the *meaning* of a representation to some other meaning. However in practice, we will have to implement our elaborations as syntactic machinations on our representations. This can become difficult when the mapping between the elements of the language and actual meaning (semantics) of the representation is either unclear or too complicated, as it will be harder to ensure that the change we make to our symbolic representation has the desired effects.

   We can perceive the meaning as being partly determined by the relationship between the elements of the representation, and our consequence relation $\mid\!\sim$. [3] To give some intuition, we can view the operation of $\mid\!\sim$ on R as the way elements of R interact to form new elements, a concept we discuss in more detail in Section 3.3.

3.3 Understanding Elaboration Tolerance

We can evaluate elaborations in terms of the following framework. Say our representation R is composed of some set of objects $\{\sigma_1, \ldots, \sigma_n\}$, each of which denotes some meaning. The inference relation $\mid\!\sim$ can be construed as combining these basic meanings to infer other meanings: we can view the set of consequences of R,

[2] We could use *Kolmogorov complexity* [Kolmogorov, 1965] to formally characterize how complicated our elaborations are, but we doubt this metric will give us any insights into our study of elaboration tolerance. On the other hand, if there was an analog of Kolmogorov complexity expressible within the language of representation, along with some set-theoretic operations, we might have a more illuminating measure.

[3] $\mid\!\sim$ could be monotonic or nonmonotonic; we make no assumptions about it yet.

$\{\phi \mid \sigma_1, \ldots, \sigma_n \mathrel{|\!\sim} \phi\}$ as the result of all sorts of "reactions" between each σ_i. (It is helpful to think of these reactions as logical steps such as modus ponens, but they could involve more than two operands.) The more consequences $\mathrel{|\!\sim}$ can derive, the more powerful it is, but this also means that it involves that many more reactions between the basic units $\{\sigma_i\}$, as well as their descendants. In this section, we demonstrate how the complexity of these reactions will in turn make it more difficult to predict how a change to $\{\sigma_1, \ldots, \sigma_n\}$ will affect its consequences.

As mentioned before, these reactions are non-existent if $\mathrel{|\!\sim}$ were simply defined to be set inclusion. If $R \mathrel{|\!\sim} \sigma \iff \sigma \in R$, then no new "products" are formed – the only conclusions we can make are those already part of R. On the other hand, we could have very complicated reactions if $\mathrel{|\!\sim}$ obeyed a rule like Cut:

$$\textit{Cut} \qquad \frac{R \mathrel{|\!\sim} \phi,\ R, \phi \mathrel{|\!\sim} \psi}{R \mathrel{|\!\sim} \psi}, \tag{3.8}$$

because now we have to deal with the reactions between the *descendants* of $\sigma_1, \ldots, \sigma_n$ in our calculations. For example, if ϕ is some new conclusion generated from R, we have to consider as conclusions all by-products ψ that could be generated from the expanded representation R plus ϕ.

Now consider what would happen to our representation if we applied the simplest elaboration, altering just one of the basic units of meaning:

$$e \cdot R = \{\sigma_1, \ldots, e(\sigma_i), \ldots, \sigma_n\}. \tag{3.9}$$

If $\mathrel{|\!\sim}$ is powerful, it will be hard to predict how this change will effect the consequences of the elaborated representation $e \cdot R$, given all of the subsequent reactions between our units of meaning. We would like the consequences of $e \cdot R$ to be a simple function of those of R, and e separately. However whether we can decompose the meaning of our elaborations in this manner depends strongly on e, as shown in (3.10).

$$\begin{aligned} e \cdot R &= \{\phi \mid \{\sigma_1, \ldots, e(\sigma_i), \ldots, \sigma_n\} \mathrel{|\!\sim} \phi\} \\ &\overset{?}{=} h(e, \{\phi \mid \{\sigma_1, \ldots, \sigma_i, \ldots, \sigma_n\} \mathrel{|\!\sim} \phi\}) \end{aligned} \tag{3.10}$$

Adding elaborations, defined as when $e(\cdot) = \cdot \cup \sigma_e$, is just as complicated, but the study of inference relations has been primed for this kind of modality on representations, and less so for others. Also, semantically, addition is easy to understand, particularly if $\mid\!\sim$ is monotonic. Deletion, by contrast, is difficult to predict as it is very sensitive to the syntactic nature of the original representation. For example, in the theory

$$\alpha, (\alpha \implies \beta), \beta, \tag{3.11}$$

deleting β will not in fact remove β from our knowledge base, because it can be re-derived from the other two conjuncts (a reaction!). In order to have our elaborations work properly, we have to understand how they inter-operate with our inference relation and the units of meaning, which could be arbitrarily complicated. Other elaborations which actually involve syntactic changes to σ_i will depend on the reactions between the σs, requiring particularly intricate reasoning.

Our construction indicates that the power of $\mid\!\sim$ is the barrier to having more expressive elaborations. If our inference relation minimized reactions between our units of meaning σ_i (e.g., if they were inert!), or more generally, if the reactions were globally well-understood, then we could better understand how our elaborations, no matter how complicated, actually effect our representation. This in turn would make it easy to find elaborations which accomplish exactly the change we desire.

Many formalisms that are structured so that their underlying units are inferentially inert come to mind. Relational databases consist of sets of tuples which do not react in any way at all. In this case, the inference relation $R \mid\!\sim_{rel\ db} (\exists\overline{y})\phi(\overline{x},\overline{y})$ is simply reduced to checking if $\phi(\overline{x},\overline{z}) \in R$, for some $\overline{z}$, spiced up with some boolean combinations. Frame systems, those structures built up from frames, slots, slot values, type and cardinality constraints, etc., are relatively inert, as the only interactions in that paradigm are inheritance between classes, and constraint checking. In this case $R \mid\!\sim_{frames} \phi$ if ϕ holds for the current class, or any superclass, given that it satisfies the constraints.

In the STRIPS [Fikes and Nilsson, 1971] formalism, states of the world are represented by sets of arbitrary first-order sentences. The system employs a first-order theorem prover to infer what is true in a given state. However it uses a much simpler framework to compute the result of performing an action, using only the set and instantiation operations on the `Precondition`, `Add`, and `Delete` lists. In contrast to full first-order inference methods for computing successor states, as was done in [Green, 1969], these set operations limit the interactions between sentences, efficiently enabling an elegant class of inferences.

In some sense, this weak interaction between units of meaning can be construed as being "Cartesian." By Cartesian, we mean that each unit of meaning σ_i is independent of the other meanings $\sigma_1, \ldots, \sigma_n$ in terms of inferential capability. These ideas are very similar to [McCarthy, 1997]'s intuition about *Cartesian theories*, in the sense that each element σ of R expresses exactly one concept, and is independent of the other concepts. This Cartesian independence will make a surgical (where we actually do manipulate the formulas) approach to elaboration tolerance more feasible – if one fact is changed, we can guarantee that there will be no [unintended] propagated effects. Also, we can iteratively change the theory one meaning σ at a time, converging on the intended representation R'. If a theory was not Cartesian in this sense, we would have to perform surgery on many different statements, all while respecting the entangled relationships between them. The intuition behind finding a proper basis in linear algebra fits well here. [4]

Another way to promote this independence is through inference rules. If our units of meaning $\sigma_1, \ldots, \sigma_n$ are truly independent, then adding some new unit of meaning σ_{n+1} will not change the meanings already derived. If this were true for any set of σ_i, then $\mid\!\sim$ would be *monotonic*. Monotonic consequence relations guarantee that none of our meanings ever change or are removed as we add statements (unless we explicitly delete them). This domain is more benign than nonmonotonic inference, where a consequence could be nullified upon *adding* a new fact.

Yet a third way to restrict these reactions between units is to use *contexts* to

[4] This Cartesian notion is the motivation for *state vectors*, introduced in [McCarthy and Painter, 1967] and used to represent counterfactuals in [Costello and McCarthy, 1999]. State vectors encode independence of elements, so that if we treat an elaboration as an operation which changes one dimension of a state vector, then the operators are truly local, and therefore predictable.

partition our elements σ_i. *Contexts* [Guha, 1991] [McCarthy, 1993] [McCarthy and Buvač, 1994] can be used to group relevant elements, and localize inference, to prevent [consideration of] interactions between irrelevant elements. This also provides a mechanism to visualize inference, as the contexts provide conceptual pathways along which inference is allowed to flow. Contexts have been used as an effective mechanism in Cyc [Cyc, 2003, Lenat and Guha, 1990], both to organize axioms, and provide bounds on inference. Hierarchies have been similarly utilized to successfully segment interactions between elements of R, as done in frame systems and object-oriented programming. [Amir, 2001], inspired by object-oriented programming methods, shows how to *partition* theories in order to encapsulate them, and prevent irrelevant inferences between them.

A fourth solution is to use a very simple language, usually lacking boolean connectives and quantifiers, to represent R. In practice, this is usually some syntactically restricted fragment of first-order logic. For example, relational databases are really just collections of positive atoms. Since they are usually interpreted in terms of the *closed world assumption* (atoms not in the database are assumed to be false), each representation corresponds exactly to one model, in an obvious fashion. The popularity of action description languages [Gelfond and Lifschitz, 1993], [Giunchiglia and Lifschitz, 1998], [Kakas et al., 1999] used to solve the frame problems in the early 1990s can also attest to the attractiveness of this approach. [5]

It may be possible to also induce inertness by *reformulating* our representations to maximize irrelevance between our units of meaning. We have to be careful however

[5] It is interesting to note that these restricted approaches closely emulate relational database theory in two major ways:

1. The formulas of the language consist solely of positive atoms (no boolean operators), just as in relational databases. Some approaches allow some limited quantification over a known set, but since the equivalent grounded set is finite it does not add expressive power. The implementation of this limited quantification itself does add some expressive power, in the sense that it is implementing a kind of *closed world assumption*.

2. There is a need for both a query language (for discovering the consequences of the data) and a description language (for representing the data) [Lifschitz, 1996]. The query language corresponds to something like a relational algebra, while the description language is like a database schema.

that our reformulations preserve the meaning of what we want to say, along with any relevancies we want to retain.

Ultimately, however, we believe that the key to ensuring our syntactic manipulations have the intended semantic effects is to ensure some sort of 1-1 correspondence between symbols and meaning. This way, when we manipulate the symbols, we directly manipulate the meaning. Then all we must ensure is that we have used our symbols to properly convey the desired meaning of R. We explore some design principles that advance this 1-1 correspondence next, in Section 3.4.

3.4 Design Principles of Elaboration Tolerance

There are a number of design principles that can reinforce this 1-1 correspondence for any representation, without changing the semantics or adding any new features. The bulk of these principles have been derived from relational database design, but also hail from compiler theory and programming methodology.

The tenets of relational database design are particularly relevant to elaboration tolerance. The express purpose of databases is to provide a system that can rapidly access, manage, and update information over time. By definition, these structures embody the essence of what is required for elaboration tolerance! Compiler theory also contributes to our theory of elaboration tolerance by distinguishing how expressions are constructed. Programming methodology underscores some of the same principles espoused by relational database design.

For this section, assume our language is sorted, with sorts $S_1, \ldots, S_n$. Note that the sorts can be repeated in the definition of a relation. *Attributes* are various renamings of the sorts used for a certain purpose. So for example we can have the sort $S_{nine\ digit\ numbers}$, but then have the attributes A_{SSN} (SSN = Social Security Number) and $A_{account\ number}$ which both contain the elements of $S_{nine\ digit\ numbers}$, but play very different roles semantically. We specify relations by the cross-product of the attributes (as opposed to sorts) of which it is a subset.

In the next few sections we explore various principles and techniques that can advance our 1-1 correspondences between meanings and symbols, on various levels of the language, from terms, to literals, to sentences.

3.4.1 How to Construct Relations

Dependencies and Keys

In order to ensure a representation reflects the proper semantics, we first have to be able to describe the semantics. Relational database design uses *dependencies* to represent these semantics, which are then used to construct proper relations. If α and β are sets of attributes, the dependency $\alpha \rightarrow \beta$ means that knowing the values of α will determine the values of β within any relation. Hence for example, $A_{SSN} \rightarrow A_{Name}$.

A *key* for a relation is some set of attributes, which when known, determines the rest of the attributes in the relation. Consider the relation $R \subseteq A_{SSN} \times A_{Name} \times A_{EyeColor}$. A_{SSN} is a key with respect to this set, because once a social security number is known for a person, the name and eye color of the person can be discovered. On the other hand $A_{EyeColor}$ is not a key for the other two attributes, as there are many people with the same eye color. Even the set $\{A_{Name}, A_{EyeColor}\}$ is not a key for A_{SSN}, because there could be multiple people with the same name and eye color. On the other hand, A_{SSN} is *not* a key for the relation $R' \subseteq A_{SSN} \times A_{Address} \times A_{Height} \times A_{time}$, where now we capture the height and address of a person at a particular time. A_{SSN} is not sufficient to determine the three attributes. However the set $\{A_{SSN}, A_{time}\}$ is a key for R'.

The notions of key and dependencies can help us construct relations which express exactly one meaning. The first concept that is useful is that of *normal forms*, which instructs how to create relations based on a set of dependencies. A set of relations satisfy *Boyce-Codd normal form* (BCNF) when, for each relation R and dependency $\alpha \rightarrow \beta$, if $\alpha \cup \beta$ is a subset of the attributes of R, then α must be the key for R. This requirement forces a relation to be a function of only the attributes that are related by a key. So for example, an illegal Boyce-Codd relation under this form would be one based on a person's SSN, their address, and the price of tea in China. There is no key in this relation that will determine the price of tea in China – it is irrelevant to the other two attributes. In short, Boyce-Codd forces relations to be as small as possible by cutting out un-functional and therefore irrelevant attributes.

Independencies

Independencies have also been studied in the database literature. They are best expressed by *embedded multivalued dependencies*, written $\alpha \twoheadrightarrow \beta \mid \gamma$, for sets of attributes α, β, and γ. This rule holds for a relation R when, if any two tuples agree on their values in α, then the relation contains the cross product of the attributes in β and γ. For example, if

$$\begin{aligned} \alpha &= \{A_{SSN}\}, \\ \beta &= \{A_{Kid's\ Name}\},\ and \\ \gamma &= \{A_{Pet's\ Name}\}, \end{aligned} \tag{3.12}$$

and we have some relation R relating a person's social security number to the names of their children and pets:

$$R \subseteq A_{SSN} \times A_{Kid's\ Name} \times A_{Pet's\ Name}, \tag{3.13}$$

then if

$$\begin{aligned} &(314\text{-}15\text{-}9265, Anju, Amy) \in R, \quad and \\ &(314\text{-}15\text{-}9265, Abhis, Stripes) \in R \end{aligned} \tag{3.14}$$

then so are

$$\begin{aligned} &(314\text{-}15\text{-}9265, Anju, Stripes) \in R, \quad and \\ &(314\text{-}15\text{-}9265, Abhis, Amy) \in R \end{aligned} \tag{3.15}$$

This independency of β and γ given α is the same as used in probabilistic reasoning, corresponding to the notion of conditional independence: $I(\beta, \gamma \mid \alpha) \iff P(\beta, \gamma \mid \alpha) = P(\beta \mid \alpha) * P(\gamma \mid \alpha)$ [Pearl and Verma, 1987].

Independencies show how data is correlated through a common set of attributes. Once this common set is known, the data becomes independent of each other. This suggests that the correlated information should be stored in separate relations, each with the common attributes as a key. Otherwise, if we put them in the same relation, we would acquire many repetitions, since we are taking the "cross-product" of them. Thus, while dependencies shrink a relation to consist of only those functionally related attributes, independencies cut the relation up further, demanding that the relation only model one real relationship at a time.

3.4.2 Syntax Matters, or Reification is Important

Related to this notion of keys and (in)dependencies is syntax. As a motivating example, consider two different ways of representing a cross action for the missionaries and cannibals problem. To assert that a row action has taken place, with two rowers $cannibal1$ and $missionary2$, from $bank1$ to $bank2$ at time t_0 we write:

$$Row(cannibal1, missionary2, bank1, bank2, t_0) \tag{3.16}$$

We call this form of asserting facts *synthetic syntax*, because it describes the concept of *Row* as being built up entirely from a vector of properties, such as rowers, bank locations, etc. On the other hand, we could use a language with *analytic syntax*, which tells one how to take statements apart:

$$\begin{aligned}
&Rowing(r)\\
&Rower(r, cannibal1)\\
&Rower(r, missionary2)\\
&SourceLocation(r, bank1)\\
&TargetLocation(r, bank2)\\
&Time(r, t_0)
\end{aligned} \tag{3.17}$$

Analytic and synthetic syntax are described in [McCarthy, 1962]. We argue that the language presented in (3.17) is much more elaboration tolerant than that presented in (3.16). The synthetic approach is highly elaboration *in*tolerant, as it cannot easily handle elaborations such as changing the boat's capacity. (3.16) also cannot distinguish between different rowing actions occurring on different boats, as there is no parameter in the relation used to refer to boats.

(3.16) contains a kind of closed world assumption, in the sense that the truth of *Row* only depends on the arguments given. What is amazing is that this hard coded truth is not dependent on any complex properties of the formalism, but by the most innocuous feature – *the syntax of the language*. (3.16) stipulates that the *Row* action only depends on whether two elements of the class of missionaries participate, so there is no way to express that three missionaries might be involved as we could with (3.17). Also, the concept of a *Row* occurring is encoded as a *proposition* in the language, in direct comparison to (3.17), where it is an object. The problem with propositions is that in first-order languages, there is no way to ascribe further properties to them – they are not reified.

We contrast this closed world assumption idea with (3.17), which mentions exactly what is known, no more, and no less, while (3.16) makes many other unintended assumptions. Since (3.17) is analytic, we can keep saying more things about r, simply by ascribing another property to it. This is what makes it so elaboration tolerant.

An intriguing observation is that the difference in syntax gives rise to different $\mathcal{L}$-structures. We can see by inspection that the form of relations in (3.17) *leaves open all possibilities* which are not mentioned. It is also very amenable to adding more symbols or relations. (3.16) already makes many alternative scenarios implicitly impossible simply by its syntax.

But a review of database design gives us the fundamental reason why analytic syntax is superior to the synthetic form. The object r, the rowing action in (3.17), acts as a *key* with respect to each relation. Our analytic approach always implicitly follows Boyce-Codd normal form, as it only*ascribes* properties *to* our object r. Once we have the identity of r, all possible facts about it (when it occurred, what boat was used, etc.) are accessible! By reifying the row action (as opposed to treating it as the truth of a proposition), we have highly increased the elaboration tolerance of our language.

This reification acts as a dual to the Boyce-Codd requirement above. Boyce-Codd is predicated on R first containing a dependency $\alpha \rightarrow \beta$. Note that our syntax (3.16) trivially satisfies Boyce-Codd, since it does not include any α that could be the head of a dependency. The requirement of reification espoused by analytic syntax requires every relations to contain some sort of key, thus rendering it amenable to the Boyce-Codd constraint.

Note that our formalization in (3.17) has already decomposed any independencies. For example, we could have had a relation $RowerSourceLocation(r, cannibal1, bank1)$, but this would require much repetition, as the values of *Rower* and *SourceLocation* are independent, given r.

The superiority of analytic syntax, as opposed to synthetic syntax, explains the *Davidsonian* approach [Davidson, 1966] to representing actions used in Cyc [Cyc, 2003]. [6] In fact, this representation was chosen expressly for the purpose of maximizing elaboration tolerance [Guha, 2003].

3.4.3 Synthetic Functions, or Operator Splitting

There is a middle ground between synthetic and analytic forms. As representationally inadequate as synthetic syntax is, *computationally* it may be more appealing because of its built-in closed world assumption. Consider if we are trying to find out if $cannibal1$ participated in a row action at time t_0. In the synthetic mode we would have to look for an atom of the form

$$Row(cannibal1, *, *, *, t_0), \tag{3.18}$$

whereas in the analytic case, we would have to find some element r such that

$$Rowing(r) \wedge Rower(r, cannibal1) \wedge Time(r, t_0), \tag{3.19}$$

which requires not only lookup but also expensive joins.

One way out of the difficulties of the previous section is that we rewrite statements of the form in (3.16) as:

[6] It is interesting to note that [Davidson, 1966] addresses these same issues of synthetic versus analytic syntax (amongst others) in his formalization of actions. Davidson concludes by advocating a rather analytic form for expressing facts about actions, where actions are terms in the language, and attributes are ascribed to them.

$$r = row(cannibal1, missionary2, bank1, bank2, t_0). \tag{3.20}$$

row is a function, returning an object r denoting (reifying) the rowing, but we still have the problem of elaborating row actions which may involve different boats, different capacities, etc. But this could be handled by extra predicates:

$$\begin{gathered} has\_third\_rower(r, rower3) \\ or \\ uses(r, boat2) \end{gathered} \tag{3.21}$$

This is precisely the trick of adding *action attributes* used in Lifschitz's formalization of the missionaries and cannibals problem in [Lifschitz, 2000]. The notion behind action attributes is the same as in *operator splitting*, which is used to speed up search in [Kautz et al., 1996] and [Kautz and Selman, 1996], by reducing the number of instantiations of formulas. [7]

Using these "synthetic functions" with operator splitting does allow for some elaboration tolerance; if a function is not originally constructed to depend on a property, we can always attribute it later with statements like (3.21). The one downside, other than the fact that the epistemology is messy, [8] is that it will be difficult to *omit* dependencies. Say we do not know or care about the time associated with a row action. Then we will have to add some kind of null to the time argument position of (3.20).

We could avoid this quandary by using various synthetic functions $row1$, $row2$,

[7][Kautz and Selman, 1996] translate first-order formulas to propositional ones by *grounding* them; the atom $move(x, y, z, i)$ which represents a move of object x from y to z at time i has $O(n^4)$ instantiations, where n is the size of the domain. In contrast, the alternative representation $object(x, i) \wedge source(y, i) \wedge destination(z, i)$ has only $O(3n^2)$ instantiations. This representation makes sense when there is exactly one action allowed per time step. It is ironic that this splitting was originally introduced not to effect elaboration tolerance, but promote efficiency in search.

[8]By messy, we mean that fact that we will need "extraction" axioms to describe fields of the synthetic function. For example, for the $row1$ action shown in (3.23), we would additionally have to say:

$$\begin{gathered} Rower(row1(r1, r2, l1, l2, t), r1) \\ Rower(row1(r1, r2, l1, l2, t), r2) \\ Time(row1(r1, r2, l1, l2, t), t) \ldots, \end{gathered} \tag{3.22}$$

and so forth, while this information is already captured by statements like (3.17).

$row3, \ldots$:

$$\begin{aligned} & r = row1(r1, r2, l1, l2, t), \\ & r' = row2(r1, r2, r3, l1, l2, t), \\ & r'' = row3(r1, r2, r3, l1, l2), \\ & \ldots \end{aligned} \tag{3.23}$$

which all return different row objects based on different input specifications.

We believe that the solution lies in using synthetic notions only across those attributes that are certain to apply to all instances. For example, if time is an attribute that might be later omissible, it should not be included in the synthetic function. These guidelines however are sensitive to the future intended use of the representation.

3.4.4 The Unique Roles Assumption

Probably the design principle we believe most important to advancing elaboration tolerance is the *unique roles assumption* (URA) [Maier and Warren, 1982], a notion not advertised enough in the literature, in our opinion. It is a database design principle originally applied to attributes, asserting that each attribute should play exactly one semantic role in the database. Hence instead of using the sort $S_{nine\ digit\ numbers}$ to play both the roles of an SSN and account number, we use two separate attributes, A_{SSN} and $A_{account\ number}$ for each.

This idea can be generalized to all our symbols, in that *every symbol has a unique meaning and use in the representation.* After our discussion of the semantic nature of elaborations, we see how the unique roles assumption helps, as it forces us to use the symbols of our language in a one-to-one correspondence with the meaning.

Having unique roles applies both to terms of our language, as well as propositions. Consider the formalization of the missionaries and cannibals problem:

$$\begin{aligned} & Missionaries(3) \\ & Cannibals(3) \\ & \ldots \end{aligned} \tag{3.24}$$

In (3.24) the symbol "3" plays two different roles, both the number of missionaries, as well as the number of cannibals. To subsequently change either of these values will require some understanding of the different roles of each, which in turn will require some understanding of the formalization in (3.24). A much better approach to (3.24) is:

$$\begin{aligned}
&Missionaries(NumM)\\
&Cannibals(NumC)\\
&NumM = 3\\
&NumC = 3\\
&\ldots,
\end{aligned} \tag{3.25}$$

where $NumM$ and $NumC$ are special constants used to denote the number of missionaries and cannibals, respectively. (3.25) obeys the unique roles assumption. We see that any elaboration that requires a change in either number will simply require a change to $NumM/NumC$, in exactly one place. All the elaborator needs to know is the meanings of $NumM$ and $NumC$, and nothing else about the theory. Example 9.5.1 shows how the URA works hand in hand with additive elaboration tolerance.

[Parmar, 2002] called for some theory of "maximal parameterization," where if two quantities refer to the same value, they should employ the same parameter. This is the converse of the unique roles assumption, which we also accept. A final generalization of the URA that we espouse is that *every meaning should have a unique symbol/name*. For example, in software engineering, macros and global variables are use to abbreviate quantities which play special roles. Programmers purposefully distinguish certain quantities as variables/macros, fully expecting them to be later modified to contain different values.

This concept of unique roles also applies to propositions. Just as atoms embody some unit of meaning, an expression built from these atoms has its own, more complicated, meaning. To extend elaboration tolerance beyond relational databases, we need to be able to name each of these more complicated concepts, and ensure that they always play a unique role.

Consider the following fragment of a formalization, adapted from [Amir, 2000]:

$$\begin{aligned}
&0 < Num(Missionaries, location, t) < Num(Cannibals, location, t) \implies \\
&\qquad Eaten(location, t) \\
&0 < Num(Missionaries, location, t) \wedge At(BigCannibal, location, t) \implies \\
&\qquad Eaten(location, t) \\
&\ldots
\end{aligned} \tag{3.26}$$

The first axiom of (3.26) asserts that when the missionaries are outnumbered on a bank by the cannibals, someone will be eaten. The second says that if the *BigCannibal*, who is large enough to bully all of the missionaries, is co-located with a missionary, someone will be eaten. (This is an adaption of elaboration nine from [McCarthy, 1997].)

Now consider what happens if we want to add an elaboration about "cannibal food." Specifically, the missionaries will not be eaten if there is some cannibal food in the area (elaboration eighteen). This would require us to change (3.26) to:

$$\begin{aligned}
&0 < Num(Missionaries, location, t) < Num(Cannibals, location, t) \implies \\
&\qquad Eaten(location, t) \vee \mathbf{At}(\mathbf{CannibalFood}, \mathbf{location}, \mathbf{t}) \\
&0 < Num(Missionaries, location, t) \wedge At(BigCannibal, location, t) \implies \\
&\qquad Eaten(location, t) \vee \mathbf{At}(\mathbf{CannibalFood}, \mathbf{location}, \mathbf{t}) \\
&\ldots
\end{aligned} \tag{3.27}$$

This adjustment to (3.26) requires too much brain surgery. We have to find each of the axioms pertaining to being eaten. Then, we must understand the purpose of each of the axioms, and then know that inserting $\vee At(CannibalFood, location, t)$ into the consequents will give us the updated theory.

Instead, we should have recognized that the preconditions $0 < Num(Missionaries, location, t) < Num(Cannibals, location, t)$ and $0 < Num(Missionaries, location, t) \wedge At(BigCannibal, location, t)$ play special roles in describing *potentially dangerous conditions* for the missionaries:

$$
\begin{aligned}
&0 < Num(Missionaries, location, t) < Num(Cannibals, location, t) \implies \\
&\qquad \mathbf{Dangerous}(\mathbf{location}, \mathbf{t}) \\
&0 < Num(Missionaries, location, t) \wedge At(BigCannibal, location, t) \implies \\
&\qquad \mathbf{Dangerous}(\mathbf{location}, \mathbf{t}) \\
&\mathbf{Dangerous}(\mathbf{location}, \mathbf{t}) \implies Eaten(location, t) \\
&\ldots
\end{aligned}
\tag{3.28}
$$

Then, in order to elaborate about cannibal food, we only require the one change:

$$
\begin{aligned}
&0 < Num(Missionaries, location, t) < Num(Cannibals, location, t) \implies \\
&\qquad Dangerous(location, t) \\
&0 < Num(Missionaries, location, t) \wedge At(BigCannibal, location, t) \implies \\
&\qquad Dangerous(location, t) \\
&Dangerous(location, t) \implies Eaten(location, t) \vee \mathbf{At}(\mathbf{CannibalFoor}, \mathbf{location}, \mathbf{t}) \\
&\ldots
\end{aligned}
\tag{3.29}
$$

Now not only is our tinkering of (3.28) restricted to only one axiom, we are also confident that we have correctly implemented the desired elaboration.

Example 9.5.1 shows how these above mentioned principles can promote additive elaboration tolerance.

As a final observation, it is interesting to note that the sole solution to the frame problem that does not employ some sort of additive solution plus nonmonotonicity is the successor state axiom approach [Reiter, 1991] and [McIlraith, 2000], which encodes a solution to inertia in an axiom of the form

$$
\begin{aligned}
&Poss(a,s) \implies \\
&\qquad [F(x, result(a,s)) \iff \gamma_F^+(x,a,s) \vee \nu_F^+(x, result(a,s)) \vee \\
&\qquad\qquad [F(x,s) \wedge \neg\gamma_F^-(x,a,s) \wedge \neg\nu_F^-(x, result(a,s))]],
\end{aligned}
\tag{3.30}
$$

where $Poss(a,s)$ is a symbol meant to represent when an action is possible. $\gamma_F^+(x,a,s)$ abbreviates the conditions that make $F(x,\cdot)$ true in $result(a,s)$, while $\gamma_F^-(x,a,s)$ are those which can make it false. $\nu_F^+(x, result(a,s))$ is comprised of the static constraints which cause $F(x,\cdot)$ to be true in $result(a,s)$ while $\nu_F^-(x, result(a,s))$ are those which make it false. $Poss(a,s)$ is already used as a way to reify the action preconditions, and although not part of the formal language, the Greek symbols are used in practice to reify the fluent preconditions. In fact, a formalism that used the successor state axiom approach, along with statements of the following form separately defining these symbols

$$
\begin{aligned}
&\gamma_F^+(x,a,s) \equiv_{def} \ldots \\
&\gamma_F^-(x,a,s) \equiv_{def} \ldots \\
&\nu_F^+(x,s) \equiv_{def} \ldots \\
&\nu_F^-(x,s) \equiv_{def} \ldots
\end{aligned}
\tag{3.31}
$$

would make the action formalism much more elaboration tolerant, as each of the modes for how an action changes a fluent are explicitly named.

Part II

Additive Elaboration Tolerance

Chapter 4

Additive Elaboration Tolerance

In this chapter we introduce the notion of *additive elaboration tolerance*, where elaborations are effected simply by adding the proper formulas to the axioms of our representation. After some justifications for this modality in Section 4.1, we describe our conceptual groundwork in Section 4.2, some of which is an extension of the framework of hard and soft facts of Section 3.1. Section 4.3 walks through our conception of two kinds of elaborations: full retraction, and full addition. Section 4.4 demonstrates that a propositional system can be outfitted to have these two kinds of additive elaborations. This system is the blueprint for how we shall endow arbitrary systems with elaboration tolerance. It also allows us to surmise some necessary properties of such a system, described in Section 4.5.

4.1 Why Additive Elaboration Tolerance?

We have championed additive elaboration tolerance, where our representations are altered simply by conjoining the proper formulas. As discussed in Chapter 1, there are three main reasons for following this approach:

1. Additive elaborations approximate human discourse.

2. Changing meanings by adding formulas allows us to push the semantics of the elaboration into the target language itself, so that the elaboration itself can be

reasoned about. This is the essence of the declarative approach to AI.

3. It minimizes any brain surgery we might have to perform on our representation – all we do is find the formula corresponding to the meaning of our elaboration, and add it to our current representation, letting the semantics take care of the rest. The elaborator is required to know little about how the representation is actually formalized, resulting in a nice *abstraction barrier*.

4.2 Tenets and Groundwork

Informally, our study of additive elaboration relies on the following assumptions and insights, some repeated from Section 3.1:

1. The only way to alter our representation R is by adding formulas such as ψ_e which specify the change in question. Formally speaking, our elaborations are operations of the form:

$$e \cdot R = \psi_e(R) \cup R = R' \tag{4.1}$$

   We can imagine a representation undergoing a sequence of elaborations by continually adding more formulas, from R to $\psi_e \cup R$ to $\psi_{e'} \cup \psi_e \cup R$, and so on.

2. Our only means of "accessing" R is by means of consequence relations – in other words, we only know what follows from our representation R using $\vdash$ and $\mid\!\sim$.

3. We use exactly two consequence relations, $\vdash$ and $\mid\!\sim$. $\vdash$ is meant to infer the hard truths from R, while $\mid\!\sim$ is used to determine softer truths.

4. Facts derivable by $\vdash$ are hard, and must always hold with respect to our elaborations. Therefore, they cannot be altered by elaborations. Since our elaborations are accomplished by adding formulas, this means $\vdash$ must be monotonic. This fits in well with our notion that hard facts are not subject to revision or retraction under any circumstances. In terms of our symbols, we can say:

$$R \vdash \phi_{hard} \implies \psi_{e_n} \cup \ldots \cup \psi_{e_1} \cup R \vdash \phi_{hard}, \tag{4.2}$$

for any sequence of elaborations $\psi_{e_1}, \ldots, \psi_{e_n}$.

Recall that $\vdash$ surveys truth across *all* possible worlds. If ϕ is a hard truth, then it holds in every world we can imagine. If $\vdash$ is monotonic, then adding formulas to its left hand side strictly decreases (by set inclusion) the set of worlds it checks for truth. Hence the set of hard truths can only increase in this paradigm of additive elaborations.

5. The soft facts inferred by $\mid\!\sim$ are based on informal arguments using uncertain or defeasible information. In terms of possible worlds, this kind of information restricts inference to some set of preferred worlds, a subset of all the worlds. Since these soft facts can be uninferred upon learning new information, $\mid\!\sim$ will have to be nonmonotonic. Hence there are elaborations $\psi_{e_1}, \ldots, \psi_{e_n}$ such that

$$\begin{gathered} R \mid\!\sim \phi_{soft}, \\ \textit{but} \\ \psi_{e_n} \cup \ldots \cup \psi_{e_1} \cup R \not\mid\!\sim \phi_{soft}, \end{gathered} \tag{4.3}$$

We add in one final assumption, that

6. *Our elaborations work by selecting a different set of preferred worlds for a representation R.*

This notion was already illustrated in Figure 3.2.

Within this framework, we can represent any elaboration that can be represented in human discourse, as we are assuming our system is strong enough to handle meta-statements about the language within the language. If R is meant to represent the original missionaries and cannibals problem, and $\psi_{3\to4}(R)$ is the elaboration that changes the numbers of missionaries and cannibals each from three to four, then $\psi_{3\to4}(R) \cup R$ should represent the new problem, which turns out to be unsolvable. In terms of the illustration in Figure 3.2, the original set of preferred models of R

are those in which the numbers of missionaries and cannibals are both three. Adding $\psi_{3\to4}(R)$ to R pushes the focus of preferred models to another part of the space, where there are four missionaries and four cannibals, as is shown below in Figure 4.1.

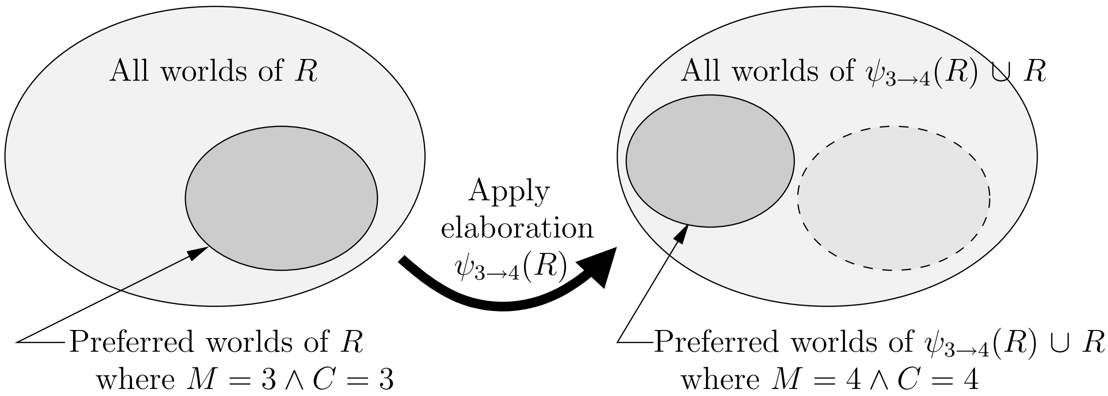

Figure 4.1: Adding $\psi_{3\to4}(R)$ to R results in switching the set of preferred worlds where the numbers of missionaries and cannibals are both four.

In terms of our framework, the fact that there were three missionaries and three cannibals must have been soft, or defeasible. The reason why is because we could choose another set of preferred models where in fact there are four of each set. It may seem contradictory that this fact is soft, because it was explicitly mentioned in the description of R, and appears to be a hard truth. This contradiction arises because of the flexibility of human language, where any utterance can be retracted or changed at a later time, without contradiction. The semantics of human discourse are such that *every expressible statement* is defeasible, and therefore in our terms a soft fact. Hence when R corresponds to some representation in a human language, the set of all worlds describing R is essentially all worlds, as almost every statement that is uttered is a soft fact (i.e. subject to retraction). [1]

In this thesis we take this insight to heart, and use $\mid\!\sim$ exclusively as our method for inference, as it deals in the realm of what is *likely*, which is how any inference that efficiently deals with the real world must work.[2] In this context, $\vdash$ acts as the dual

[1] This apparent contradiction buttresses our agnostic view of the actual representation of R, and the use of $\vdash$ and $\mid\!\sim$ to inspect it indirectly.

[2] [Costello, 1997] notes that "In life, as in the jungle, waiting until you are absolutely sure that the large striped animal is indeed a tiger is often fatal."

modality to $\mid\!\sim$, only determining what facts are *necessary*. Or, conversely, we can use $\vdash$ to determine what facts are *possibly true*, by the formula $R \not\vdash \neg\phi$ – it is not the case that in all worlds $\neg\phi$ holds, which means there is a world of R where ϕ holds.

A brief aside: Notice that we cast our elaborative formula ψ to be a function of the current representation R, as in $\psi_{3\to4}(R)$. Had we not allowed the dependence, the commutativity of $\cup$ would force some contradictions, such as that:

$$\psi_{4\to3} \cup \psi_{3\to4} \cup R = \psi_{3\to4} \cup \psi_{4\to3} \cup R. \tag{4.4}$$

The left hand side of (4.4) should be the original missionaries and cannibals problem with three of each, while the right hand side seems to represent the problem with four of each. Explicitly recording the dependence of ψ on the representation to which it is applied avoids this problem, shown in (4.5).

$$\psi_{4\to3}(\psi_{3\to4}(R) \cup R) \cup \psi_{3\to4}(R) \cup R \neq \psi_{3\to4}(\psi_{4\to3}(R) \cup R) \cup \psi_{4\to3}(R) \cup R. \tag{4.5}$$

In the sequel, we will often drop the dependence of our elaborations on the representation, and let the order in which formulas are conjoined indicate the order in which they were applied. We will assume $\cup$ is right associative, so that

$$\psi_{e_3} \cup \psi_{e_2} \cup \psi_{e_1} \cup R = \psi_{e_3} \cup (\psi_{e_2} \cup (\psi_{e_1} \cup R)) \tag{4.6}$$

4.3 Retraction and Addition of Formulas

We have defined additive elaboration tolerance in terms of adding the proper elaborative formulas to our representation R. In this section we make our first steps in studying how the semantics of R can implement our desired elaborations, *within* the language. To do so, we formalize two major classes of additive elaborations: retracting and adding formulas. Once we have these means of essentially adding and subtracting statements from our representation, representing arbitrary elaborations boils down to whether they can be expressed in the base representation R. If a formalism cannot represent some elaboration, then by definition the elaboration will have to be implemented by altering the syntactic form – exactly the kinds of operations we wish to avoid. In fact, [McCarthy, 1998] observes that "elaborations not expressible as additions to the object language representation may be treatable as additions at a meta-level expression of the facts." We want to maximize the first category, and minimize the second.

We discuss retraction before addition, because the definition of addition will depend in part on retraction.

4.3.1 Retractable Formulas

First of all, we want to be able to *retract formulas by adding the appropriate elaboration*. By this we mean, given any α that is a soft (defeasible) fact of R, we want to be able to add some sentence $\psi_\emptyset(\alpha, R)$ to R resulting in a representation which is, by default, agnostic about α. This idea seems paradoxical at first, until we note our $\psi_\emptyset(\alpha, R)$ is equivalent to the English statement "forget α in R."

Of course, this property of full retraction by definition only applies to the set of soft facts, since it is impossible to ever retract a hard fact. (There are no alternative worlds to go to where the hard fact does not hold.)

In other words, we want $\psi_\emptyset(\alpha, R)$ to obey:

$$\psi_\emptyset(\alpha, R) \cup R \not\mid\!\sim \alpha \tag{4.7}$$

But this kind of retractability is easy to satisfy, if we have negation ($\neg$) in our language, as we will generally assume. To accomplish (4.7), simply set $\psi_\emptyset(\alpha, R)$ to $\neg\alpha$. This is clearly not what we want, because instead of being agnostic about α, we will infer it does not hold. If a person is told to "forget that Tweety flies," he does not instead infer in retribution that Tweety does not fly. In fact, he believes that either case is possible, and he would not be surprised if either fact became later known to be true.

Hence, we need to come up with a stronger definition of what we mean by retractability to accommodate negation. Intuitively, if we do not want to know either α or $\neg\alpha$, in terms of possible worlds we want our $\psi_\emptyset(\alpha, R)$ to switch the focus of the current set of preferred worlds, to some other set where both α and $\neg\alpha$ holds. If we loosely interpret $R \mid\!\sim \alpha$ as meaning that α holds in all preferred worlds of R, then we want:

$$\begin{aligned} &\psi_\emptyset(\alpha, R) \cup R \not\mid\!\sim \alpha \wedge \\ &\psi_\emptyset(\alpha, R) \cup R \not\mid\!\sim \neg\alpha, \end{aligned} \tag{4.8}$$

as this connotes the meaning that "there is a preferred world of $\psi_\emptyset(\alpha, R) \cup R$ where α does not hold, and there is a preferred world of $\psi_\emptyset(\alpha, R) \cup R$ where $\neg\alpha$ does not hold." Syntactically, this also makes sense, as after absorbing elaboration $\psi_\emptyset(\alpha, R)$, we cannot softly infer *whether* α holds or not.

This notion of retractability is still not adequate for our purposes. We mentioned earlier that we plan to add sequences of elaborations to our representation, and a proper definition of retractability should allow for us to retract a formula after *any* intervening sequence of elaborations. This notion of being retractable in the context of *any* elaboration we call *full retractability.* If $\Psi_\mathcal{E}$ is some class of elaborations closed under sequence, we can say α is fully retractable in R if it satisfies:

$$(\forall \Psi \in \Psi_\mathcal{E})[(\exists \psi_\emptyset \in \Psi_\mathcal{E})[(\psi_\emptyset \cup \Psi \cup R \not\mid\!\sim \alpha) \wedge (\psi_\emptyset \cup \Psi \cup R \not\mid\!\sim \neg\alpha)]] \tag{4.9}$$

(4.9) indicates that no matter what elaborations from some class $\Psi_\mathcal{E}$ have been added to R, we can always find some other formula $\psi_\emptyset$ to persuade us to be agnostic about α.

Unfortunately, this definition is not quite correct. If we have an elaboration of the form $\psi_!(\alpha)$ in $\Psi_\mathcal{E}$ establishing a hard truth, akin to asserting "α is true and can never become false, no matter what is ever subsequently said about it," then full retractability will never be true for α. In order to allow for full retractability within the presence of these formulas we must restrict our definition:

α is *fully retractable* if:

$$\begin{aligned}&(\forall\Psi \in \Psi_{\mathcal{E}})[(\Psi \cup \Gamma \nvdash \alpha) \wedge (\Psi \cup \Gamma \nvdash \neg\alpha) \implies \\ &(\exists\psi_\emptyset \in \Psi_{\mathcal{E}})[(\psi_\emptyset \cup \Psi \cup R \not\mathrel{|}\!\sim \alpha) \wedge (\psi_\emptyset \cup \Psi \cup R \not\mathrel{|}\!\sim \neg\alpha)]].\end{aligned} \tag{4.10}$$

This is the definition of full retractability that we will employ throughout this thesis. Our precondition that $(\Psi \cup R \nvdash \alpha) \wedge (\Psi \cup R \nvdash \neg\alpha)$ means that Ψ forces neither α, nor $\neg\alpha$, to hold in the elaborated representation $\Psi \cup R$. In other words, there is no formal proof in $\Psi \cup R$ to show either case. If this is true, then α is fully retractable when we can add a formula $\psi_\emptyset$ to block even any informal argument that could lead to concluding the truth of α. Of course α is *trivially fully retractable* if $(\Psi \cup \Gamma \vdash \alpha) \vee (\Psi \cup \Gamma \vdash \neg\alpha)$ for those $\Psi \in \Psi_{\mathcal{E}}$.

We can also interpret (4.10) in terms of worlds. Say there are worlds of $\Psi \cup R$ where α holds and worlds where $\neg\alpha$ holds. Then α is fully retractable if we can find some formula $\psi_\emptyset$ which when added to $\Psi \cup R$, will change the focus of preferred models to include both models where α, and its negation holds. The precondition just reassures us that there are such models that can be included in our new set of preferred worlds. The intuition is illustrated in Figure 4.2.

Figure 4.2 shows how as we add elaborations, our set of *all* models gets smaller since $\vdash$ is monotonic. It appears possible to “run out” of models, particularly when our representation has only finitely many formulas. It is reasonable to assume that R requires some minimal level of complexity in order to admit full retraction. We discuss this further in Section 4.5, while Theorem 5.6.3 proves that the minimal level of complexity required is infinitely many symbols in the language, given a typical $\mathrel{|}\!\sim$.

4.3.2 Addable Formulas

In this section we investigate the notion of adding facts to our representation R. At first this idea seems unnecessary – if one wants to conclude α, simply add α to the representation. However in practice it may not be as simple as that. α may directly contradict R, so that the result, $\alpha \cup R$ may be *inconsistent*. Even worse, we may later decide that we want to retract α, and depending on our semantics, it may not

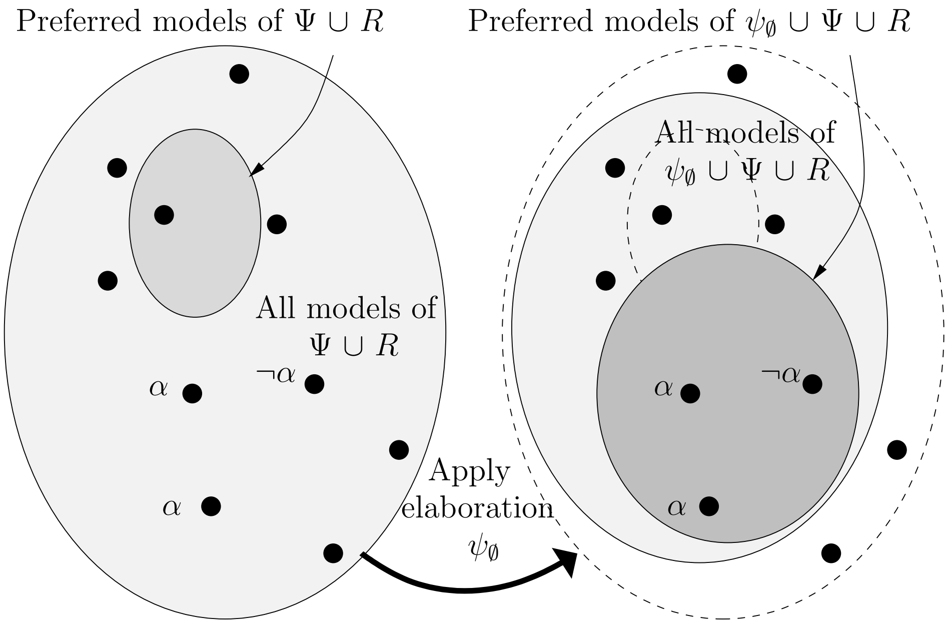

Figure 4.2: A graphical depiction of full retraction. The dots correspond to models, and the shaded ellipses pick out sets of models. Notice how the set of *all* models of $\Psi \cup R$ becomes smaller with the addition of $\psi_\emptyset$, while the preferred ones simply change.

be possible to do so.

To sidestep these problems, we instead introduce a formula $\psi_+(\alpha, R)$ which does the adding for us. It obeys:

$$\begin{aligned} &\psi_+(\alpha, R) \cup R \mid\!\sim \alpha \wedge \\ &Consis(\psi_+(\alpha, R), R) \wedge \\ &\mathit{fully\text{-}retractable}(\alpha, \psi_+(\alpha, R) \cup R) \wedge \\ &Consis(\neg\alpha, R) \implies Consis(\neg\alpha, \psi_+(\alpha, R) \cup R) \end{aligned} \tag{4.11}$$

We need all four conjuncts in (4.11), because it is not enough for $\psi_+(\alpha, R)$ to force R to conclude α; the combination $\psi_+(\alpha, R) \cup R$ must be consistent as well, as otherwise it would be enough to set $\psi_+(\alpha, R)$ to $\perp$, and thus conclude any formula.

We define consistency in terms of $\vdash$ in (4.12). α is consistent in R if R does not formally prove $\neg\alpha$. Semantically speaking, this means there is a world of R where α

holds.

$$Consis(\alpha, R) \equiv_{def} R \not\vdash \neg\alpha \tag{4.12}$$

We also want α to be fully retractable in the new representation $\psi_+(\alpha, R) \cup R$. By this we mean we want the freedom to step back from any of our declarations if we later desire. To have this property, we need to know that $\neg\alpha$ was preserved in at least one world, through the addition of $\psi_+(\alpha, R) \cup R$. These last two conditions taken together prevent our $\psi_+(\alpha, R)$ in question from being set to our proposed $\psi_!(\alpha, R)$, which permanently asserts α.

We are almost done refining our notion of full addition. As with our definition of full retraction, we want to be able to force R to consistently conclude α, even after some arbitrary sequence of elaborations from some set $\Psi_{\mathcal{E}}$ are added to R. Hence we write:

$$\begin{aligned}(\forall\Psi \in \Psi_{\mathcal{E}})(\exists\psi_+ \in \Psi_{\mathcal{E}})[&Consis(\psi_+, \Psi \cup R) \wedge \\ &(\psi_+ \cup \Psi \cup R \mid\!\sim \alpha) \wedge \\ &\mathit{fully\text{-}retractable}(\alpha, \psi_+ \cup \Psi \cup R) \wedge \\ &Consis(\neg\alpha, \Psi \cup R) \implies Consis(\neg\alpha, \psi_+(\alpha, R) \cup \Psi \cup R)\end{aligned} \tag{4.13}$$

This formulation of what we call *full addition* is close to what we desire. What we have neglected to include is that our $\Psi \in \Psi_{\mathcal{E}}$ could be the dreaded $\psi_!(\neg\alpha, R)$, which means that adding $\psi_+(\alpha)$ would be impossible. As we did for full retractability, we will have to restrict our definition of full addition to only certain configurations of $\Psi \cup R$ where it is *consistent* to have α hold. In fact, we say α is *fully addable* if

$$
\begin{aligned}
&(\forall \Psi \in \Psi_{\mathcal{E}})[Consis(\alpha, \Psi \cup R) \implies \\
&\ (\exists \psi_+ \in \Psi_{\mathcal{E}})[Consis(\psi_+, \Psi \cup R) \wedge \\
&\qquad\qquad (\psi_+ \cup \Psi \cup R \mathrel{|\!\sim} \alpha) \wedge \\
&\qquad\qquad \textit{fully-retractable}(\alpha, \langle \mathcal{L},\ \vdash,\ \mathrel{|\!\sim},\ \psi_+ \cup \Psi \cup R\rangle) \wedge \\
&\qquad\qquad Consis(\neg\alpha, \Psi \cup R) \implies Consis(\neg\alpha, \psi_+(\alpha, R) \cup \Psi \cup R)]]
\end{aligned}
\tag{4.14}
$$

We can interpret this definition as follows. Say Ψ is an elaboration that when applied to R, does not formally derive $\neg\alpha$. Then we can find some elaboration $\psi_+(\alpha, \Psi \cup R)$ which, when applied to $\Psi \cup R$ will result in a consistent representation, that will by default infer α. Furthermore, α can be retracted (if it is possible to do so), no matter what other subsequent elaborations have been added to $\Psi \cup R$. Again, we say α is *trivially fully addable* for those $\Psi \in \Psi_{\mathcal{E}}$ such that $\neg Consis(\alpha, \Psi \cup R)$, or $\Psi \cup R \vdash \neg\alpha$.

The semantic explanation behind this definition is also intuitive. As long as our elaboration Ψ of R results in a representation which contains a world where α holds, then we can find a $\psi_+(\alpha, \Psi \cup R)$ to add to $\Psi \cup R$ to switch the set of preferred models to include only those where α holds. And yet, we can always subsequently find some other retraction formula $\psi_\emptyset(\alpha)$ to push the set of preferred worlds to another focus where α is unknowable. This intuition is illustrated in Figure 4.3.

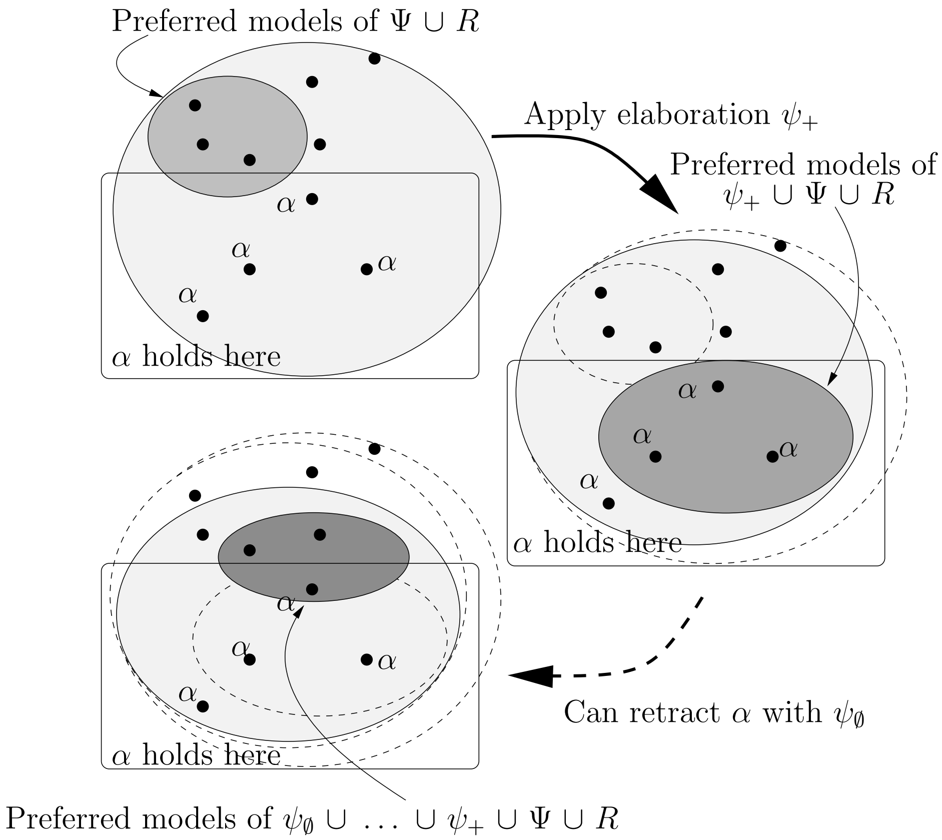

Figure 4.3: A graphical depiction of full addition. The dots correspond to worlds, and the shaded ellipses pick out sets of worlds. Provided there is a world describing $\Psi \cup R$ where α holds, we can add ψ_+ to our axioms, to change the set of preferred worlds to lie entirely within the extent of $Mod(\alpha)$. Furthermore, this set of models is non-empty. The second dotted transformation illustrates that we can always later move the set of preferred models out of $Mod(\alpha)$, if there is a world remaining where $\neg\alpha$ holds.

4.4 A Simple System with Full Retraction and Full Addition

In this section, we provide a small taste of how we plan to construct elaboration tolerant representations from those which are not. Let $Prop$ be some finite set of propositional symbols, and let $\mathcal{L}_{Prop}$ be the propositional language built from these atoms. This section shows how to augment an arbitrary propositional $\mathcal{L}_{Prop}$-theory Γ so that it is fully retractable and fully addable for any formula within the original language $\mathcal{L}_{Prop}$. This construction should give the reader a proper understanding of the workings of the general constructions which are to follow, as the fundamental principles are the same.

Our construction is straightforward: from our $\mathcal{L}_{Prop}$-theory Γ, we construct the abnormalized version Γ_{Ab}, which is simply the set of statements $Ab(n) \vee \gamma$ for each $\gamma \in \Gamma$. Ab is a new unary predicate and n a natural number, different for each γ. We let $\vdash$ be the usual classical inference relation over our new language $\mathcal{L}_{Prop} \cup \{Ab\} \cup \mathbb{N}$. Since every $\mathcal{L}_{Prop}$ formula is prefixed with $Ab(n)$ in Γ_{Ab}, we see that there are no hard non-tautological truths of Γ_{Ab} in the language $\mathcal{L}_{Prop}$.

Our defeasible inference relation $\mid\!\sim$ is just a version of circumscription where the extension of Ab is minimized, and all other symbols are varied.[3] In other words, $\mid\!\sim$ will assert a fact iff it holds in all Ab-minimal models. Formally, we can define:

$$\Delta \mid\!\sim \phi \equiv_{def} Circ[\Delta[Ab]; Ab; Prop] \vdash \phi \tag{4.15}$$

Since every sentence in Γ_{Ab} is prefixed with an Ab, the circumscription provides a natural pressure to assume as many $\mathcal{L}_{Prop}$-propositions hold as possible, without falling into inconsistency.

This natural pressure also provides us with a means to construct our $\psi_{\emptyset}(\gamma)$ and $\psi_{+}(\gamma)$. Intuitively, the Ab in each sentence $Ab(n) \vee \gamma$ is a label which can be used as a "gate" to turn off γ, when so desired, as nicely observed in [McDermott, 1987]. In order to retract γ from our set of conclusions, we simply add the atom $Ab(n)$ to the sentences in Γ_{Ab}, and then let circumscription take care of the rest.

This is the idea, although in this simple form, will require too much brain surgery by our standards. Say Γ_{Ab} contains the set of statements

[3]If $Prop$ were not finite, our standard circumscription formula would be infinitary. In the general case, this will not matter, as we will use a semantic version of model preference.

$$\begin{aligned}
&Ab(n_1) \vee \gamma_1,\\
&Ab(n_2) \vee (\gamma_1 \implies \gamma_2),\\
&\dots,\\
&Ab(n_i) \vee (\gamma_{i-1} \implies \gamma_i),
\end{aligned} \tag{4.16}$$

and we want to retract γ_i. Then we will first have to look for *each* of these deduction chains in Γ_{Ab}, and then decide where to "cut" them by asserting $Ab(n_j)$, for some j between 1 and i. Furthermore, in this framework it is not clear how to achieve full addition, that is, add a statement to force our set of axioms to conclude γ. We could in fact just add γ to our axioms, but then we could never retract it.

There is a much more elegant solution to retract γ besides meticulously asserting *Ab*s. The trick is to "destabilize" Γ_{Ab}. As we mentioned before, the circumscriptive aspect of $\mid\!\sim$ provides a natural pressure to push the *Ab*s to be false, and therefore each accompanying γ to be true. Consider what happens if we add the statements $Ab(n^+) \vee \gamma$ and $Ab(n^-) \vee \neg\gamma$, where n^+ and n^- are numbers not mentioned in the rest of the theory Γ_{Ab}. Then, the pressure of circumscription will be forced to choose between making $Ab(n^+)$ true and making $Ab(n^-)$ true. Since n^+ and n^- do not appear anywhere else in the theory, and neither γ nor $\neg\gamma$ is a hard fact, the theory cannot express a preference over which model to choose. Hence it will choose both: there will be incomparable *Ab*-minimal models both where γ holds and models where $\neg\gamma$ holds.

Thus we can define our $\psi_\emptyset$ to be:

$$\psi_\emptyset(\gamma) = (Ab(n^+) \vee \gamma) \wedge (Ab(n^-) \vee \neg\gamma) \tag{4.17}$$

Intuitively, our $\psi_\emptyset(\gamma)$ "carves" out the space of models as shown in Figure 4.4.

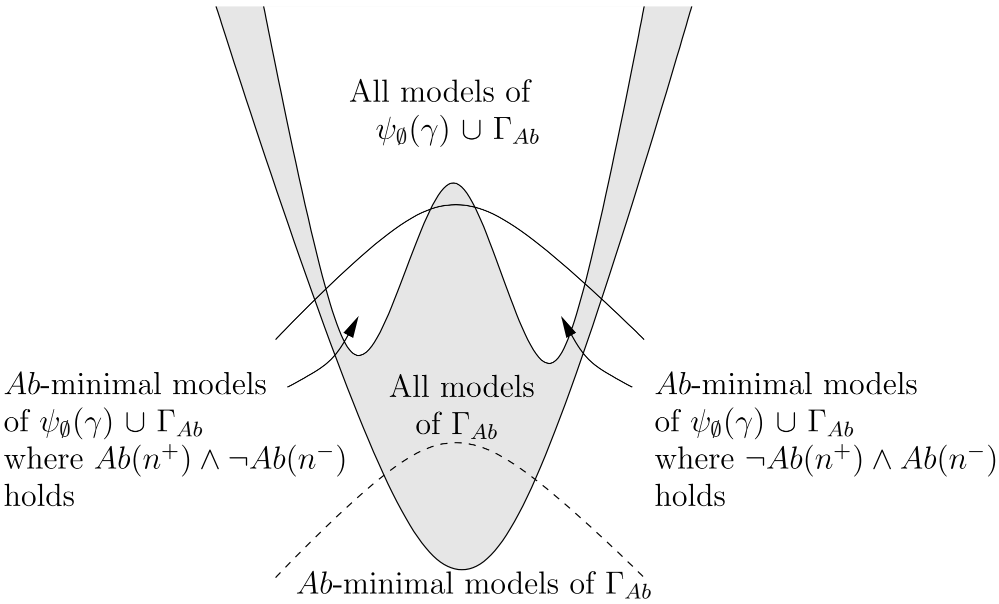

Figure 4.4: How $\psi_\emptyset(\gamma)$ carves out the space of minimal models. The gray area represents the models of Γ_{Ab}, with the Ab-minimal models toward the bottom. The white region represents the models of $\psi_\emptyset(\gamma) \cup \Gamma_{Ab}$. Note how it generates two incomparable sets of Ab-minimal models, one where $Ab(n^+) \wedge \neg Ab(n^-)$ holds, and the other where $\neg Ab(n^+) \wedge Ab(n^-)$ holds.

We will formally prove that $\psi_\emptyset$ implements full retraction in Chapter 8 when we similarly augment arbitrary systems. For now we trust $\psi_\emptyset$ properly does its job with respect to $\mathcal{L}_{Prop}$, and instead pause to extol its virtues. Compared to picking and setting Abs to retract a formula, adding $\psi_\emptyset$ is much simpler. It requires no computation, nor consideration of the form of the theory Γ_{Ab}, save for which labels have already been used, a very minor consideration. $\psi_\emptyset$ by its construction satisfies [Shanahan, 1997]'s requirement that its size be on order of the information being added. Also, it turns out $\psi_\emptyset$ works *regardless of what sequence of other elaborations from $\Psi_\mathcal{E}$ have been applied to* Γ_{Ab}. This is actually quite a strong statement, as it turns out $\Psi_\mathcal{E}$ can contain all elaborations which describe retraction and addition of formulas.

For now we provide a simple example showing how retraction works:

Example 4.4.1 (Retraction of Formulas in $\mathcal{L}_{Prop}$). Consider the $\mathcal{L}_{Prop}$-theory Γ:

$$\begin{aligned}
&\alpha \\
&\alpha \implies \beta
\end{aligned} \tag{4.18}$$

Then our abnormalized Γ_{Ab} is:

$$\begin{aligned}
&Ab(1) \vee \alpha \\
&Ab(2) \vee (\alpha \implies \beta)
\end{aligned} \tag{4.19}$$

In all Ab-minimal models, it is easy to see that α and $\alpha \implies \beta$, and therefore β holds:

$$\begin{aligned}
Ab(1) \vee \alpha, Ab(2) \vee (\alpha \implies \beta) & \mathrel{|\!\sim} \alpha, \\
& \mathrel{|\!\sim} \alpha \implies \beta, \\
& \mathrel{|\!\sim} \beta
\end{aligned} \tag{4.20}$$

Let us say we want to retract β. There are classical models of (4.19) where β and $\neg\beta$ hold, or equivalently, $\Gamma_{Ab} \not\mathrel{|\!\sim} \beta \wedge \Gamma_{Ab} \not\mathrel{|\!\sim} \neg\beta$. Hence we can consistently add

$$\begin{aligned}
\psi_{\emptyset}(\beta) \quad = \quad & (Ab(3) \vee \beta) \wedge \\
& (Ab(4) \vee \neg\beta)
\end{aligned} \tag{4.21}$$

to the sentences in (4.19). The Ab-minimal models of $Ab(1) \vee \alpha, Ab(2) \vee (\alpha \implies \beta), Ab(3) \vee \beta, Ab(4) \vee \neg\beta$ can be divided into three classes entailing the three formulas:

$$\begin{aligned}
&Ab(4) \wedge \alpha \wedge \beta \\
&Ab(2) \wedge Ab(3) \wedge \alpha \wedge \neg\beta \\
&Ab(1) \wedge Ab(3) \wedge \neg\alpha \wedge \neg\beta
\end{aligned} \tag{4.22}$$

From (4.22), we can see that neither β nor $\neg\beta$ holds in the resulting Ab-minimal models – β has been retracted! Also, we can see how the models correspond to the minimal change required to retract β. The first class retracts $\neg\beta$, allowing for the models where $\alpha \wedge \beta$ hold. The second instead decides to retract both the implication $\alpha \implies \beta$ and the fact that β holds, resulting in $\alpha \wedge \neg\beta$. Finally the third removes the assertions α and β, resulting in $\neg\alpha \wedge \neg\beta$. As mentioned before, the form of the theory matters little in the construction of our $\psi_{\emptyset}$, although it may affect what the minimal models are.

There also turns out to be an elegant solution for adding formulas, if we leverage the fact that our abnormalities use numbers ($\mathbb{N}$) to label the statements. If we want to force our representation to softly conclude γ, we could add $Ab(n) \vee \gamma$, where n is again some number not mentioned in Γ_{Ab}. Then we would hope that the circumscriptive pressure would force us to conclude γ, as asserting $Ab(n)$ unnecessarily is discouraged. However this is not enough – there could be other statements in Γ_{Ab} directly contradicting γ, which would result in the twin valleys shown in Figure 4.4. Namely, Γ_{Ab} could just be $Ab(n') \vee \neg\gamma$. By the arguments for retraction, $Ab(n) \vee \gamma, Ab(n') \vee \neg\gamma \not\models \gamma$.

Not only must we promote γ by some formula like $Ab(n) \vee \gamma$, we must also actively discourage any competing formulas which might conclude the opposite. This requirement reverts to the previous problem of having to search for and break deductive chains of formulas which entail, in this case, $\neg\gamma$. We can avail ourselves of this complication by taking advantage of the ordering on our numbers.

Say in fact we choose n in $Ab(n) \vee \gamma$ to be larger than all other numbers mentioned in Γ_{Ab}. Now consider a model of $Ab(n) \vee \gamma, \Gamma_{Ab}$ where in fact $\neg\gamma$ holds. $Ab(n)$ will have to hold in this model, but we must penalize this structure further, to ensure it cannot be an Ab-minimal model. The way to do so is to add another statement of the form $(\forall x)[Ab(n) \wedge x < n \implies Ab(x)]$. Since n is larger than all other numbers mentioned in Γ_{Ab}, any structure which entails $\neg\gamma$, and therefore $Ab(n)$, will be drastically penalized by having all Abs with parameters less than n be set to $\top$.

It turns out, as proved in Chapter 8, these two statements are enough to guarantee full addition, and hence correspond to our $\psi_+(\gamma)$:

$$\psi_+(\gamma) = (Ab(n) \vee \gamma) \wedge (\forall x)[Ab(n) \wedge x < n \implies Ab(x)], \tag{4.23}$$

where quantification is meant to be over the natural numbers. The fact that $\psi_+(\gamma)$ implements full addition is also proved in Chapter 8. For now, we pause to admire ψ_+ a bit. First, its construction only depends on the highest number mentioned in Γ_{Ab}. Since it implements full addition, this means we can apply any other sequence of elaborations from $\Psi_{\mathcal{E}}$ (which will be shown to include retractions and additions of formulas) to Γ_{Ab}, and still get the desired result. What is most interesting, however, is to look at what ψ_+ *means*. Intuitively, the first conjunct threatens to assert $Ab(n)$ if γ is not true in a structure. But threat is dwarfed by the second conjunct, which asserts that when $Ab(n)$ holds, *all* other Abs less than it must necessarily be set to $\top$, a much larger penalty than $Ab(n)$. In a loose sense, if structures want to minimize their extension of Ab, they would do well to assert γ rather than $\neg\gamma$

We conclude this section with two more examples.

Example 4.4.2 (Addition of Formulas in $\mathcal{L}_{Prop}$). Consider the theory Γ_{Ab}:

$$\begin{aligned} &Ab(1) \vee \alpha \\ &Ab(2) \vee \beta \\ &Ab(3) \vee (\alpha \wedge \beta \implies \gamma) \end{aligned} \tag{4.24}$$

The Ab-minimal models of this theory entail α, β, and γ. Say we wanted to add the conclusion $\neg\gamma$. Our updated set of axioms are:

$$\begin{aligned} &Ab(1) \vee \alpha \\ &Ab(2) \vee \beta \\ &Ab(3) \vee (\alpha \wedge \beta \implies \gamma) \\ &Ab(4) \vee \neg\gamma \\ &(\forall x)[Ab(4) \wedge x < 4 \implies Ab(x)] \end{aligned} \tag{4.25}$$

The Ab-minimal models in this case consist of three types:

$$\begin{aligned} &Ab(3) \wedge \alpha \wedge \beta \wedge \neg\gamma \\ &Ab(2) \wedge \alpha \wedge \neg\beta \wedge \neg\gamma \\ &Ab(1) \wedge \neg\alpha \wedge \beta \wedge \neg\gamma \end{aligned} \tag{4.26}$$

Notice that $\neg\gamma$ holds in all models! Also notice how we again only retract to the models which correspond to the minimal changes we would have to make to our database in order to accommodate $\neg\gamma$. In some sense, this is a *cautious* kind of reasoning, because we do not prefer one eventuality over another.

Example 4.4.3 (Continuation of Example 4.4.2). Say we change our mind, and prefer γ to hold instead. Our theory consists of

$$\begin{aligned}
&Ab(1) \vee \alpha \\
&Ab(2) \vee \beta \\
&Ab(3) \vee (\alpha \wedge \beta \implies \gamma) \\
&Ab(4) \vee \neg\gamma \\
&(\forall x)[Ab(4) \wedge x < 4 \implies Ab(x)]
\end{aligned} \tag{4.27}$$

Currently, this theory softly entails $\neg\gamma$. What happens when we add $\psi_+(\gamma) = (Ab(5) \vee \gamma) \wedge (\forall x)[Ab(5) \wedge x < 5 \implies Ab(x)]$?

After wading through the mathematics, we end up with four kinds of models of the form:

$$\begin{aligned}
&Ab(1), Ab(2), Ab(3), Ab(4), \alpha, \beta, \gamma \\
&Ab(1), Ab(2), Ab(3), Ab(4), \alpha, \neg\beta, \gamma \\
&Ab(1), Ab(2), Ab(3), Ab(4), \neg\alpha, \beta, \gamma \\
&Ab(1), Ab(2), Ab(3), Ab(4), \neg\alpha, \neg\beta, \gamma
\end{aligned} \tag{4.28}$$

Once again, γ holds, but one worry is that we have lost information about α and β, as in the face of this new elaboration, we would naturally assume them to be true again. But then on the other hand, this may make sense – if a person asserts γ and then $\neg\gamma$, this blatant about-face should make us not believe any of their assertions up to that point. This phenomenon is similar to what happens in the belief revision system of [Boutilier, 1996], except that only the facts caught in the intervening contradictory sequence are lost.

4.5 Properties of Additive Elaborative Tolerance

Figures 4.2 and 4.3 illustrate how our set of preferred worlds ebbs and flows with regard to the models of α and $\neg\alpha$ as directed by our two elaborative formulas $\psi_\emptyset(\alpha, R)$ and $\psi_+(\alpha, R)$. The set of worlds used by the monotonic $\vdash$ steadily becomes smaller and smaller. On the other hand, those picked out by nonmonotonic $\mid\!\sim$ changes more freely.

These two properties of full retractability and full addition require our R to have some special properties. If we want to be able to keep elaborating our representations by adding formulas indefinitely, it seems there must be enough – infinitely many – models of our representations, so that we never "run out" of worlds in which to ascribe truth. Hence it is unlikely, for example, that R can be represented as a propositional logic with finitely many atoms. Theorem 5.6.3 shows this fact formally, that in fact, a system that has non-trivial full retraction and addition will require the use of infinitely many formulas to do the retractions and additions.

It is also impossible within this framework for our consequence relation $\mid\!\sim$ to be monotonic. We have indicated some assumption of nonmonotonicity throughout this chapter, but formally show this in Corollary 5.6.1.

These results may seem unintuitive, because in practice we use finite languages to express our motivations. The source of the infinity comes from the fact that we can arbitrarily retract and add formulas, as many times as we desire. Hence we can imagine that we can have full retraction and addition over a finite language, as long as there are infinitely many symbols left in the background to implement our retractions and additions. Intuitively these infinitely many formulas are the fodder to provide these infinitely many models. In the example in Section 4.4, the infinity is provided by the labels $n \in \mathbb{N}$. These numerical labels allow us to timestamp our assertions.

Another way to view the situation is that our language already contains infinitely many symbols, but our only means of access, the inference relations $\vdash$ and $\mid\!\sim$, can only discern formulas built from some smaller (finite) set of symbols. This fits in with the notions of incomplete information and possible worlds.

The view of having a larger language within which to express our elaborations additively is not new. We argue that mingling our meta level declarations with the object language is necessary if we are to have powerful enough elaborations. Approaches such as belief revision which divorce the meta data (epistemic entrenchment) from the data itself will always be inadequate.

We will formally show how to add symbols to give a representation R full retraction and full addition in Chapter 7. This chapter will also demonstrate what other formal requirements R must satisfy to have these nice properties. But first we present some formal infrastructure in Chapters 5 and 6.

Chapter 5

Extended Axiomatic Formal Systems

In this thesis we use what we call *extended axiomatic formal systems* to model our representations. This is an extension of the *axiomatic formal systems* $\langle \mathcal{L}, \vdash, \Gamma \rangle$ used in [Giunchiglia and Walsh, 1992] to model *abstractions*.

In Chapter 3, we introduced the concept of hard and soft consequences $\vdash$ and $\mid\!\sim$. Section 5.1 wraps these concepts, along with information about a representation's language $\mathcal{L}$ and set of core axioms Γ, into a bundle, which we call an *extended axiomatic formal system* of the form $\langle \mathcal{L}, \vdash, \mid\!\sim, \Gamma \rangle$. This is our means of generally characterizing representations. Section 5.2 provides an accompanying intuitive semantic characterization.

After this, we consider those extended axiomatic formal systems whose semantics can be characterized in terms of *choice functions* in Section 5.3. Choice functions are one way to abstractly characterize semantics for some consequence relations. We call this class of representations *extended axiomatic formal systems with choice* (Section 5.4). We copy the exposition of choice functions used in [Lehmann, 2001]. The great benefit of this characterization is that it provides a nice framework in which we can combine two such systems to produce another system, retaining the desirable properties of its parents, as will be shown in Chapter 6.

But before doing so, we pause to construct formal definitions of full retraction and full addition in Section 5.5, and then formally prove some ramifications of a system with these properties, in Section 5.6.

5.1 Extended Axiomatic Formal Systems

Definition 5.1.1 (Extended Axiomatic Formal Systems). An *extended axiomatic formal system* is a four-tuple of the form:

$$\langle \mathcal{L},\ \vdash,\ \mid\!\sim,\ \Gamma \rangle, \tag{5.1}$$

where $\mathcal{L}$ is some language, $\vdash$ and $\mid\!\sim$ consequence relations, and Γ some $\mathcal{L}$-theory. As for $\vdash$ and $\mid\!\sim$, there is only one formal restriction, that of *supraclassicality*:

$$\textit{Supraclassicality} \quad \frac{\Gamma \vdash \alpha}{\Gamma \mid\!\sim \alpha} \tag{5.2}$$

Supraclassicality ensures that every hard consequence is included as a soft consequence. Semantically, this reinforces our notion that every preferred possible world of Γ is also a possible world of Γ. As mentioned in Chapter 4, we will usually treat $\vdash$ as some monotonic, classical inference relation, while $\mid\!\sim$ is nonmonotonic. $\mid\!\sim$ must be nonmonotonic, as *additive elaboration tolerance* requires we effect our elaborations simply by adding formulas to the left hand side of $\mid\!\sim$.

5.2 The Semantics of Extended Axiomatic Formal Systems

Let $\mathcal{L}$ be a language. [Lehmann, 2001] gives some theorems showing when a consequence relation $\mid\!\sim\ \subseteq 2^{\mathcal{L}} \times \mathcal{L}$ satisfying certain properties has an accompanying semantics in terms of choice functions. These theorems give us an alternative way of viewing how our consequence relation works, providing further insights. If $\mathcal{M}$ is some non-empty set, and $\models\ \subseteq\ \mathcal{M} \times \mathcal{L}$, then we can define the models of an $\mathcal{L}$-theory Γ as $Mod(\Gamma)$:

$$Mod(\Gamma) =_{def} \{m \in \mathcal{M} \mid (\forall \phi \in \Gamma)[m \models \phi]\} \tag{5.3}$$

We can go in the other direction and show how a set of models $X \subseteq \mathcal{M}$ relate to $\mathcal{L}$-formulas:

$$Th(X) =_{def} \{\phi \in \mathcal{L} \mid (\forall m \in X)[m \models \phi]\} \tag{5.4}$$

For a given consequence relation $\vdash \subseteq 2^{\mathcal{L}} \times \mathcal{L}$, Alfred Tarski has shown (and [Lehmann, 2001] gives a version of this proof) that for any $\mathcal{L}$-theory Γ, there are a set of models $\mathcal{M}$ and satisfaction relation $\models$ such that

$$\Gamma \vdash \phi \iff \phi \in Th(Mod(\Gamma)), \tag{5.5}$$

provided $\vdash$ satisfies the conditions of inclusion, cut, and monotony, which are consequence rules defined below in (5.6). This representation theorem (and others like it) are important because they give conditions under which we can find a set of semantics ($\mathcal{M}$ and $\models$) which can underlie $\vdash/\mid\!\sim$.

$$\begin{array}{rc} \textbf{Inclusion} & \Gamma, \phi \vdash \phi \\ \textbf{Cut} & \dfrac{\Gamma \vdash \phi, \quad \Gamma, \phi \vdash \psi}{\Gamma \vdash \psi} \\ \textbf{Monotony} & \dfrac{\Gamma \vdash \phi}{\Gamma, \psi \vdash \phi} \end{array} \tag{5.6}$$

5.3 Choice Functions

We can generalize the framework of Sections 3.1 and 4.2 to talk about an intuitive class of consequence relations that are determined semantically. Consider a *choice function* $f : 2^{\mathcal{M}} \rightarrow 2^{\mathcal{M}}$. What f does is, given a set of models, picks out the ones which are best, or most preferred. We can then define our consequence relation $\mid\!\sim$ on an $\mathcal{L}$-theory Γ as:

$$\Gamma \mid\!\sim \phi \ \equiv_{def} \phi \in Th(\mathbf{f}(Mod(\Gamma))) \tag{5.7}$$

which can be equivalently expressed as:

$$\Gamma \mid\!\sim \phi \equiv_{def} \mathbf{f}(Mod(\Gamma)) \subseteq Mod(\phi) \tag{5.8}$$

What this definition means is that given a theory Γ, we first take its models, pick out the preferred or "best" models (using f), and then find out what formulas hold in this restricted (sub)set, as shown in Figure 5.1.

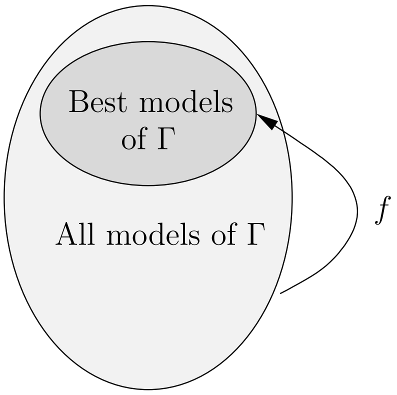

Figure 5.1: f chooses the best models of Γ.

This kind of operator meshes with our intuitions of how our soft facts are those that hold in some preferred or most likely set of worlds, as opposed to all of them. Our f provides an illuminating semantic counterpart to our $\mid\!\sim$. Classical inference can be represented by f being the identity function.

We include some other properties that our consequence relation may obey in (5.9), depending on the behavior of f:

$$
\begin{array}{rl}
\textbf{Inclusion} & \Gamma, \phi \mathrel{|\!\sim} \phi \\
\textbf{Monotonicity} & \Gamma \subseteq B \wedge \Gamma \mathrel{|\!\sim} \phi \implies B \mathrel{|\!\sim} \phi \\
\textbf{Right Monotonicity} & \Gamma \mathrel{|\!\sim} \phi \wedge \phi \vdash \psi \implies \Gamma \mathrel{|\!\sim} \psi \\
\textbf{Right Conjunction} & \Gamma \mathrel{|\!\sim} \phi \wedge \Gamma \mathrel{|\!\sim} \psi \implies \Gamma \mathrel{|\!\sim} \phi \wedge \psi \\
\textbf{Left Logical Equivalence} & \Gamma \equiv B \wedge \Gamma \mathrel{|\!\sim} \phi \implies B \mathrel{|\!\sim} \phi \\
\textbf{Left Disjunction} & \alpha \mathrel{|\!\sim} \phi \wedge \beta \mathrel{|\!\sim} \phi \implies \alpha \vee \beta \mathrel{|\!\sim} \phi \\
\textbf{Cautious Monotonicity} & \Gamma \mathrel{|\!\sim} \phi \wedge \Gamma \mathrel{|\!\sim} \psi \implies \Gamma, \phi \mathrel{|\!\sim} \psi
\end{array} \tag{5.9}
$$

Clearly f must satisfy some constraints in order for $\mathrel{|\!\sim}$ to make any sense. Some properties ascribed to f throughout this thesis include:

$$
\begin{array}{ll}
\textbf{Contraction} & (\forall X)[f(X) \subseteq X] \\
\textbf{Coherence} & (\forall X, Y)[X \subseteq Y \implies X \cap f(Y) \subseteq f(X)]
\end{array} \tag{5.10}
$$

Contraction and *coherence* [Chernoff, 1954], [Sen, 1970], [Moulin, 1985] are standard notions in the literature. Contraction requires f to restrict its set of "best" models to the set from which it chooses them, as we have already depicted in Figure 5.1. Coherence is a little more complicated. Essentially, if X is a subset of Y, $x \in X$, and x is one of the "best" models chosen from Y, then it should be one of the "best" models of X. [Lehmann, 2001] showcases these properties in his work.

There are other properties that we will find useful, faithfulness and ϕ-reflection:

$$
\begin{array}{ll}
\textbf{Faithfulness} & f(X) = \emptyset \implies X = \emptyset \\
\boldsymbol{\phi}\textbf{-Reflection} & Y \subseteq X \wedge \phi, \neg\phi \notin Th(f(X)) \wedge \phi, \neg\phi \notin Th(Y) \implies \\
& \phi, \neg\phi \notin Th(f(Y))
\end{array} \tag{5.11}
$$

Faithfulness is used to make sure f is not too picky – the only way it can return no models as its choice, is if it was given no models to choose to begin with. *ϕ-reflection* is used to assert that f preserves "reflections" of ϕ in the models: If ϕ is indeterminate in $f(X)$, as well as in $Y \subseteq X$, then it should stay so in $f(Y)$. The intuitive justification for reflection is the following. Given some facts, say you cannot tell whether ϕ *usually* holds. (This corresponds to $\phi, \neg\phi \notin Th(f(X))$.) Then say, given some more facts about the world $Y \subseteq X$, you cannot tell whether ϕ strictly holds ($\phi, \neg\phi \notin Th(Y)$). Then you should still not be able to assert whether ϕ *usually* holds given this new information ($\phi, \neg\phi \notin Th(f(Y))$). [1]

f embodies exactly our intuitions hinted at in Chapter 4 in the sense of picking out the most likely/preferred models. If we define $\Gamma \mathrel{|\!\sim} \phi \iff f(Mod(\Gamma)) \subseteq Mod(\phi)$, and $\Gamma \vdash \phi \iff Mod(\Gamma) \subseteq Mod(\phi)$, then we can see how $\vdash$ determines what can be determined from classical consequence, while $\mathrel{|\!\sim}$ gives facts that hold in the best or most likely models of the world.

Notice that we have made no restrictions on $\mathcal{L}$ or $\mathcal{M}$, so that these concepts apply to *any* language. We conclude this section with some useful representation facts about the properties described in (5.9) and (5.10).

Proposition 5.3.1 [Properties of $\mathrel{|\!\sim}$ defined by f]. *Say* $\Gamma \mathrel{|\!\sim} \phi \iff f(Mod(\Gamma)) \subseteq Mod(\phi)$. *Then:*

1. *$\mathrel{|\!\sim}$ obeys right monotonicity and right conjunction.*

2. *If f obeys contraction then $\mathrel{|\!\sim}$ obeys inclusion.*

3. *Say $f(X)$ is defined to be of the form*

$$f(X) =_{def} \{x \in X \mid (\forall y \in X)[R(x, y)]\}, \tag{5.12}$$

for some relation r. Then f satisfies contraction, coherence, and left disjunction.

[1] If reflection were true for every $\phi \in \mathcal{L}$, [van Benthem, 2003] argues $f(Y)$ would have to be either equal to $f(X)$, or equal to Y, based on an argument that we could set exactly two worlds in $f(X)$ and Y to have ϕ true, so that $f(Y)$ must contain one of them.

4. *Say $f(X)$ is defined to be of the form*

$$f(X) =_{def} \{x \in X \mid (\forall y \in X)[\neg y < x]\} \tag{5.13}$$

where $<$ is well-founded and transitive (no infinitely descending chains). Then $\mid\!\sim$ satisfies cautious monotonicity.

Proofs are in Section A.1.

The converse, that the properties of $\mid\!\sim$ map to properties of the choice function f, requires some more restrictions. We can determine properties of f from $\mid\!\sim$ only on those sets of models which are *definable*, because $\mid\!\sim$ only operates in terms of logical theories. The notion of undefinable sets of models is not new. The class of all finite models is not definable [Enderton, 1972], as well as the class of all complete partial and linear orderings [Dickmann, 1985].

Definition 5.3.1 (Definable Sets). Let X be a set of models of $\mathcal{M}$. X is *definable* if there exists a $\Gamma \subseteq \mathcal{L}$ such that $X = Mod(\Gamma)$.

In other words, X is definable if there is an $\mathcal{L}$-theory whose models are X.

Proposition 5.3.2 [Properties of f defined by $\mid\!\sim$]. *Let $\Gamma \mid\!\sim \phi \iff f(Mod(\Gamma)) \subseteq Mod(\phi)$. Then:*

1. *Say $\mid\!\sim$ obeys Inclusion. Then f obeys contraction on the definable subsets of $\mathcal{M}$.*

Proofs are in Section A.2.

5.4 Extended Axiomatic Formal Systems with Choice

We can now define an important subclass of extended axiomatic formal systems, those whose consequence relation $\mid\!\sim$ can be defined in terms of a choice function:

Definition 5.4.1 (Extended Axiomatic Formal Systems with Choice). $\langle \mathcal{L},\ \vdash,\ \mid\!\sim,\ \Gamma\rangle$ is an *extended axiomatic formal system with choice* if there is a relation $\models\ \subseteq\ Struct(\mathcal{L}) \times \mathcal{L}$ and $f : 2^{Struct(\mathcal{L})} \rightarrow 2^{Struct(\mathcal{L})}$, such that [2]

$$\begin{array}{rcl} \Gamma \mid\!\sim \phi & \Longleftrightarrow & f(Mod(\Gamma)) \subseteq Mod(\phi) \\ & \textit{and} & \\ \Gamma \vdash \phi & \Longleftrightarrow & Mod(\Gamma) \subseteq Mod(\phi) \end{array} \tag{5.14}$$

where f satisfies the principles of contraction and coherence.

Remember that Mod and Th are defined from $\models$ by (5.3) and (5.4).

Notice that within this construction, the fact that f satisfies contraction means $\langle \mathcal{L},\ \vdash,\ \mid\!\sim,\ \Gamma\rangle$ obeys supraclassicality:

Proposition 5.4.1 [Supraclassicality]. *Let $\langle \mathcal{L},\ \vdash,\ \mid\!\sim,\ \Gamma\rangle$ be an extended axiomatic formal system with choice. Then the relations $\vdash$ and $\mid\!\sim$ obey supraclassicality.*

Proof is in Section A.3.

These extended axiomatic formal systems with choice are very useful, because we can use f as a window into the workings of $\mid\!\sim$. This will be very useful when we seek to combine two extended axiomatic formal systems with choice, and come up with reasonable semantics for the combination in Chapter 6.

5.5 Definitions of Full Retraction and Full Addition

We have informally presented definitions of full retraction and full addition in Section 4.3. In this section we formally define these concepts in terms of extended axiomatic formal systems of the form $\langle \mathcal{L},\ \vdash,\ \mid\!\sim,\ \Gamma\rangle$.

Definition 5.5.1 (Full Retraction). A formula α is *fully retractable* if we can add some formula to retract α as a soft truth, as long as it is not a hard truth with respect to any previous elaboration.

[2] $Struct(\mathcal{L})$ refers to the class of $\mathcal{L}$-structures.

$$
\begin{aligned}
&\mathit{fully\text{-}retractable}(\alpha, \langle \mathcal{L},\ \vdash,\ \mid\!\sim,\ \Gamma\rangle) \equiv_{def} \\
&\quad (\forall \Psi \in \Psi_{\mathcal{E}})[(\Psi \cup \Gamma \nvdash \alpha) \wedge (\Psi \cup \Gamma \nvdash \neg\alpha)] \implies \\
&\quad (\exists \psi_{\emptyset} \in \Psi_{\mathcal{E}})[(\psi_{\emptyset} \cup \Psi \cup \Gamma \mid\!\not\sim \alpha) \wedge (\psi_{\emptyset} \cup \Psi \cup \Gamma \mid\!\not\sim \neg\alpha)]
\end{aligned}
\tag{5.15}
$$

$\Psi_{\mathcal{E}}$ is a set consisting of all conjunctions of possible elaborative formulas that can be added to Γ. No matter what other elaborations Ψ have been conjoined to Γ, as long as they neither entail α nor $\neg\alpha$, then we can add another formula $\psi_{\emptyset}$ to ensure that neither α nor $\neg\alpha$ are soft conclusions.

Call $\psi_{\emptyset}(\alpha, \Psi, \Gamma)$ the formula which retracts α. We restrict the possible elaborations Ψ to the set $\Psi_{\mathcal{E}}$. Restricting the set of possible elaborative formulas will help us to prove properties about full retractability later. We can formally define trivial full retraction, alluded to in Section 4.3.1 as follows:

Definition 5.5.2 (Trivial Full Retraction). Let $\langle \mathcal{L},\ \vdash,\ \mid\!\sim,\ \Gamma\rangle$ be as above, and α some $\mathcal{L}$ formula. α is *trivially fully retractable* if:

$$
(\exists \Psi \in \Psi_{\mathcal{E}})[(\Psi \cup \Gamma \vdash \alpha) \vee (\Psi \cup \Gamma \vdash \neg\alpha)]
\tag{5.16}
$$

Now in the context of this powerful kind of retractability, we can formally define addition of formulas:

Definition 5.5.3 (Full Addition). A formula α is *fully addable* if we can consistently add a formula to softly conclude α, as long as α is consistent with previous elaborations. Furthermore, we can always retract α later:

$$
\begin{aligned}
&\mathit{fully\text{-}addable}(\alpha, \langle \mathcal{L},\ \vdash,\ \mid\!\sim,\ \Gamma\rangle) \equiv_{def} \\
&\quad (\forall \Psi \in \Psi_{\mathcal{E}})[\mathit{Consis}(\alpha, \Psi \cup \Gamma) \implies \\
&\quad (\exists \psi_{+} \in \Psi_{\mathcal{E}})[\mathit{Consis}(\psi_{+}, \Psi \cup \Gamma) \wedge \\
&\qquad\qquad (\psi_{+} \cup \Psi \cup \Gamma \mid\!\sim \alpha) \wedge \\
&\qquad\qquad \mathit{fully\text{-}retractable}(\alpha, \langle \mathcal{L},\ \vdash,\ \mid\!\sim,\ \psi_{+} \cup \Psi \cup \Gamma\rangle) \wedge \\
&\qquad\qquad \mathit{Consis}(\neg\alpha, \Psi \cup \Gamma) \implies \mathit{Consis}(\neg\alpha, \psi_{+} \cup \Psi \cup \Gamma)]]
\end{aligned}
\tag{5.17}
$$

Remember that $Consis(\alpha, \Gamma) \equiv \Gamma \nvdash \neg\alpha$.

Trivial full addition is when we cannot always add something to softly conclude α:

Definition 5.5.4 (Trivial Full Addition). Let $\langle \mathcal{L}, \vdash, \mid\!\sim, \Gamma \rangle$ be as above and α some $\mathcal{L}$-formula. α is *trivially fully addable* if:

$$(\exists \Psi \in \Psi_{\mathcal{E}})[\neg Consis(\alpha, \Psi \cup \Gamma)], \tag{5.18}$$

which is the same as

$$(\exists \Psi \in \Psi_{\mathcal{E}})[\Psi \cup \Gamma \vdash \neg\alpha], \tag{5.19}$$

meaning that $\neg\alpha$ is a hard fact and therefore it is impossible to add α.

This time call the existentially qualified formula above $\psi_+(\alpha, \Psi, \Gamma)$. We can ascribe some intuitive meanings to $\psi_\emptyset(\alpha, \Psi, \Gamma)$ and $\psi_+(\alpha, \Psi, \Gamma)$. $\psi_\emptyset(\alpha, \Psi, \Gamma)$ means "forget α in $\Psi \cup \Gamma$", since adding it to $\Psi \cup \Gamma$ ensures that neither α nor $\neg\alpha$ can be consequently inferred. $\psi_+(\alpha, \Psi, \Gamma)$ on the other hand, means something like "assume α for now." It has less force than "assert α" since we must be able to retract it later.

5.6 Properties of Full Retraction and Addition

Now that we have a formal definition of full retraction and full addition, we can prove formalisms with these capabilities have some interesting properties.

5.6.1 Elaborations Can Be Sequenced

First we would like to verify that Definitions 5.5.1 and 5.5.3 actually formalize the property of formulas being retractable and addable after *any* sequence of elaborations have been conjoined to our axioms. The following two theorems show that full retractability and addition of formulas is preserved as we conjoin more and more elaborations from $\Psi_{\mathcal{E}}$.

If α is fully retractable from an axiomatic system $\langle \mathcal{L},\ \vdash,\ \mid\!\sim,\ \Gamma\rangle$, it is fully retractable in any elaborated version of $\langle \mathcal{L},\ \vdash,\ \mid\!\sim,\ \Gamma\rangle$:

Theorem 5.6.1 [Full Retraction Definition is Correct]. *Let $\langle \mathcal{L},\ \vdash,\ \mid\!\sim,\ \Gamma\rangle$ be an extended axiomatic formal system. Let $\psi_1, \ldots, \psi_n$ be any sequence of formulas from $\Psi_{\mathcal{E}}$. Then*

$$\begin{aligned} &fully\text{-}retractable(\alpha, \langle \mathcal{L},\ \vdash,\ \mid\!\sim,\ \Gamma\rangle) \implies \\ &\qquad fully\text{-}retractable(\alpha, \langle \mathcal{L},\ \vdash,\ \mid\!\sim,\ \psi_n \cup\ \ldots\ \cup \psi_1 \cup \Gamma\rangle). \end{aligned} \tag{5.20}$$

Proof is in Section A.4.

If α is fully addable in an axiomatic system $\langle \mathcal{L},\ \vdash,\ \mid\!\sim,\ \Gamma\rangle$, it is fully addable in any elaborated version of $\langle \mathcal{L},\ \vdash,\ \mid\!\sim,\ \Gamma\rangle$:

Theorem 5.6.2 [Full Addition Definition is Correct]. *Let $\langle \mathcal{L},\ \vdash,\ \mid\!\sim,\ \Gamma\rangle$ be an extended axiomatic formal system. Let $\psi_1, \ldots, \psi_n$ be any sequence of formulas from $\Psi_{\mathcal{E}}$. Then*

$$\begin{aligned} &fully\text{-}addable(\alpha, \langle \mathcal{L},\ \vdash,\ \mid\!\sim,\ \Gamma\rangle) \implies \\ &\qquad fully\text{-}addable(\alpha, \langle \mathcal{L},\ \vdash,\ \mid\!\sim,\ \psi_n \cup\ \ldots\ \cup \psi_1 \cup \Gamma\rangle). \end{aligned} \tag{5.21}$$

Proof is in Section A.5.

5.6.2 The Infinitary Requirement of Full Retraction and Addition

Section 4.5 observes that a system with full retraction and addition will require infinitely many models. In this section we formally show that an infinite language (and therefore infinitely many models) is required for our version of additive elaboration tolerance.

Theorem 5.6.3 [Full Retraction and Addition Require Infinite Languages]. *Let $\langle \mathcal{L},\ \vdash,\ \mid\!\sim,\ \Gamma\rangle$ be an extended axiomatic formal system with choice which obeys* *faithfulness and left logical equivalence, and say there is a formula* $\alpha \in \mathcal{L}$*, which is fully retractable and fully addable in* Γ*, and not trivially so, for either case.*

Then $\mathcal{L}$ *must have infinitely many formulas.*

Proof is in Section A.6.

The essence of this proof is based on alternately retracting and asserting some formula $\alpha \in \mathcal{L}$. If there are not infinitely many elaborative formulas, the retraction/addition formulas will have to repeat at some point. But then, because of left logical equivalence our elaborated retraction will be equivalent to the previous addition, so that either the retraction or addition must fail. Note that left logical equivalence rules $\mid\!\sim$ out as some sort of resource-based consequence relation, such as those used in linear logics. [3]

Corollary 5.6.1 [Nonmonotonicity of $\mid\!\sim$]. $\mid\!\sim$*, for the conditions in Theorem 5.6.3 must be nonmonotonic.*

Proof is in Section A.7.

[3] Linear logics, with their emphasis on resources, could play an intriguing alternate testbed for studying additive elaboration tolerance.

Chapter 6

Combining Extended Axiomatic Formal Systems

In this chapter we demonstrate a way to combine an extended axiomatic formal system with choice $S_1 = \langle \mathcal{L}_1,\ \vdash_1,\ \mid\!\sim_1,\ \Gamma_1 \rangle$ by another system $S_2 = \langle \mathcal{L}_2,\ \vdash_2,\ \mid\!\sim_2,\ \Gamma_2 \rangle$. Our motivation is to study ways to combine two systems to result in a third one which shares the elaboration tolerant features of its parents. This combination is a cleaner way to endow a system with elaboration tolerance, which itself is rather elaboration tolerant.

Since they are such a clean framework, it is easy to construct simple but intuitive combinations of extended axiomatic formal systems with choice. [Gabbay, 1996], [Gabbay, 1992], and other similar works already show how to combine various logical systems to create new ones. [Gabbay, 1992] shows one way to combine nonmonotonic systems. In this section we propose a different method that will be utilized in later chapters. Section 6.1 shows how to construct a mixed language from $\mathcal{L}_1$ and $\mathcal{L}_2$, while Section 6.2 shows how to make sense of it all. Finally Section 6.3 shows how the updated hard and soft consequence relations $\vdash_{1,2}$ and $\mid\!\sim_{1,2}$ are constructed.

6.1 Combining Languages

Let $S_1 = \langle \mathcal{L}_1, \vdash_1, \mid\!\sim_1, \Gamma_1 \rangle$ and $S_2 = \langle \mathcal{L}_2, \vdash_2, \mid\!\sim_2, \Gamma_2 \rangle$ be two extended axiomatic systems with choice. Let f be the choice function for $\mid\!\sim_1$, and g that for $\mid\!\sim_2$. Assume that $\mathcal{L}_1$ and $\mathcal{L}_2$ are at most first-order. If there are quantifiers in the language, distinguish between quantification over the two structures as $\forall_1$ and $\forall_2$. We would like to create some kind of combined system $S_2 \cdot S_1 = \langle \mathcal{L}_{1,2}, \vdash_{1,2}, \mid\!\sim_{1,2}, \Gamma_{1,2} \rangle$, which weaves the semantics of both systems.

First of all, we create a mixed language $\mathcal{L}_{1,2}$:

$$\mathcal{L}_{1,2} \quad =_{def} \quad Closure(\mathcal{L}_1 \cup \mathcal{L}_2), \tag{6.1}$$

where $Closure$ combines the symbols of $\mathcal{L}_1$ and $\mathcal{L}_2$ under boolean operators from either language (if there are any). The one restriction is that if the languages are first-order with equality, then no "sharing" occurs of terms with predicates. For example, we do not allow mixed atoms of the form $t_1 = t_2$ or $P(t_1, t_2)$, where t_1 is a term of $\mathcal{L}_1$ and t_2 a term of $\mathcal{L}_2$. The two languages are assumed to be mutually exclusive. [1]

Given this language $\mathcal{L}_{1,2}$, our new set of axioms $\Gamma_{1,2}$ can be some combination of Γ_1 and Γ_2, along with new formulas of $\mathcal{L}_{1,2}$.

6.2 Combining Semantics

In order to describe how $\vdash_{1,2}$ and $\mid\!\sim_{1,2}$ work, we need to ascertain satisfaction in this mixed language $\mathcal{L}_{1,2}$. For this we develop Cartesian products of structures; if $Struct(\mathcal{L}_1)$ and $Struct(\mathcal{L}_2)$ are the classes of $\mathcal{L}_1/\mathcal{L}_2$-structures, then our new set of $\mathcal{L}_{1,2}$-structures is just the cross-product $Struct(\mathcal{L}_1) \times Struct(\mathcal{L}_2)$. We can then define satisfaction of $\mathcal{L}_{1,2}$ structures on $\mathcal{L}_{1,2}$ formulas:

Definition 6.2.1 (Satisfaction ($\models_{1,2}$) of $\mathcal{L}_{1,2}$-structures). Let ϕ be a formula of $\mathcal{L}_{1,2}$ and $(m_1, m_2) \in Struct(\mathcal{L}_1) \times Struct(\mathcal{L}_2)$, so that m_1 is an $\mathcal{L}_1$-structure and m_2

[1] If we wanted to truly mix the languages, say by allowing non-logical operators from $\mathcal{L}_1$ to operate over formulas of $\mathcal{L}_2$ and vice versa, we would have to employ the more complicated *fibred semantics* described in various publications including [Gabbay, 1996], [Gabbay, 1992] and [Gabbay and Nossum, 1997]. This simpler kind of mix is sufficient for our purposes.

an $\mathcal{L}_2$-structure. Call $\models_1$ the satisfaction relation for S_1 and $\models_2$ the one for S_2. We define $\models_{1,2} \subseteq (\mathit{Struct}(\mathcal{L}_1) \times \mathit{Struct}(\mathcal{L}_2)) \times \mathcal{L}_{1,2}$ as:

$$(m_1, m_2) \models_{1,2} \phi \equiv_{def} \begin{cases} m_i \models_i \phi & \phi \in \mathcal{L}_i \\ [(m_1, m_2) \models_i \alpha] \mathbin{\#} [(m_1, m_2) \models_i \beta] & \phi = \alpha \mathbin{\#} \beta \\ & (\#\text{ a boolean connective}) \\ (\forall a \in |m_i|)[(m_1, m_2) \models_i \psi[x/a]] & \phi = (\forall_i x)[\psi] \end{cases} \tag{6.2}$$

Essentially, each model decides the truth of its own $\mathcal{L}_i$-formula, broken down by any boolean connectives. $\forall_i$ refers to quantification in m_i's domain, so that the two universes of the two models are kept separate, and quantification in each language still ranges over the intended universes.

The index on the satisfaction relations is simply to remind us of that we are operating in system i.

With a definition for $\models_{1,2}$ in hand, we can construct definitions of $Mod_{1,2}$ and $Th_{1,2}$ a lá (5.3) and (5.4).

Now we extend our choice functions f and g from the domains of $\mathit{Struct}(\mathcal{L}_1)$ and $\mathit{Struct}(\mathcal{L}_2)$, respectively, to $\mathit{Struct}(\mathcal{L}_1) \times \mathit{Struct}(\mathcal{L}_2)$. To do so, we first introduce some simplifying notation:

Definition 6.2.2 ($[A]_\ell$ and $[A]_r$).

$$\begin{aligned} [A]_\ell &=_{def} \{a \mid (a, b) \in A\} \\ [A]_r &=_{def} \{b \mid (a, b) \in A\} \end{aligned} \tag{6.3}$$

$[\cdot]_\ell$ and $[\cdot]_r$ is just shorthand for stripping off certain sides of the tuples in A. It is like the *project* relation used in relational databases.

Definition 6.2.3 (Extending Choice Functions). Say $f : 2^{\mathit{Struct}(\mathcal{L}_1)} \to 2^{\mathit{Struct}(\mathcal{L}_1)}$ and $g : 2^{\mathit{Struct}(\mathcal{L}_2)} \to 2^{\mathit{Struct}(\mathcal{L}_2)}$. We can construct extended versions f^* and g^* to cover the domain $2^{\mathit{Struct}(\mathcal{L}_1) \times \mathit{Struct}(\mathcal{L}_2)}$ into $2^{\mathit{Struct}(\mathcal{L}_1) \times \mathit{Struct}(\mathcal{L}_2)}$, as shown in (6.4). Say $W \subseteq \mathit{Struct}(\mathcal{L}_1) \times \mathit{Struct}(\mathcal{L}_2)$:

$$\begin{aligned} f^*(W) &=_{def} \{(m_1, m_2) \in W \mid m_1 \in f([W]_\ell)\} \\ g^*(W) &=_{def} \{(m_1, m_2) \in W \mid m_2 \in g([W]_r)\} \end{aligned} \tag{6.4}$$

Note that we cannot assume f^* and g^* commute. But there is some interesting structure present in these choice functions. We include some relevant properties of our extensions f^* and g^* in Proposition 6.2.1.

Proposition 6.2.1 [Consequences of f^* and g^*]. *Let f^* and g^* be as in Definition 6.2.3. then:*

1. *f^* and g^* satisfy contraction.*
2. *If f satisfies coherence, so does f^*. The same goes for g^*.*
3. *f^* satisfies* right interchangeability, *defined as:*

$$(a, b) \in f^*(w) \wedge (a, c) \in w \implies (a, c) \in f^*(w). \tag{6.5}$$

   g^* satisfies left interchangeability.

4. *If f satisfies faithfulness, so does f^*.*
5. *If f satisfies ϕ-reflection for $\phi \in \mathcal{L}_1$, so does f^*.*

Proof is in Section A.8.

6.3 The Combined System $S_{1,2} = S_2 \cdot S_1$

With these functions, we can now finally define our consequence relations $\vdash_{1,2} \subseteq 2^{\mathcal{L}_{1,2}} \times \mathcal{L}_{1,2}$ and $\mid\!\sim_{1,2} \subseteq 2^{\mathcal{L}_{1,2}} \times \mathcal{L}_{1,2}$:

$$\begin{aligned} A \vdash_{1,2} \phi &\equiv_{def} f^*(Mod(A)) \subseteq Mod(\phi) \\ A \mid\!\sim_{1,2} \phi &\equiv_{def} f^*(g^*(Mod(A))) \subseteq Mod(\phi) \end{aligned} \tag{6.6}$$

Notice that we have defined our hard truths $\vdash_{1,2}$ in our new system $S_{1,2} = \langle \mathcal{L}_{1,2}, \vdash_{1,2}, \mid\!\sim_{1,2}, \Gamma_{1,2} \rangle$ as the soft consequence relation $\mid\!\sim_1$ of S_1, while $\mid\!\sim_{1,2}$ is the composition of f^* and g^*. Our "hard" reasoning in $S_{1,2}$ involves looking at what holds in all f^*-preferred models. On the other hand, default reasoning corresponds first to seeing what g^* deems more likely, and then based on that, using f^* to pick out the worlds deemed likeliest of those. The idea is that g^* constructs a set of soft truths *relative to the hard ones of* f.

Earlier we motivated our use of $\vdash$ versus $\mid\!\sim$ in terms of $\vdash$ being monotonic while $\mid\!\sim$ was not. For the combined system $S_{1,2}$ whether $\vdash_{1,2}$ is monotonic, will depend on f and g. For our purposes it will be enough that the consequences of $\mid\!\sim_{1,2}$ include those of $\vdash_{1,2}$; that is that $\mid\!\sim_{1,2}$ is soft *relative to* $\vdash_{1,2}$.

Chapter 7

The Combined System $S_{1,Ab}$

In this chapter we show how to formally endow an arbitrary extended axiomatic system with choice $S_1 = \langle \mathcal{L}_1,\ \vdash_1,\ \mid\!\sim_1,\ \Gamma_1 \rangle$ with full retraction and full addition for any formula in its language $\mathcal{L}_1$. We do so by combining S_1 with a special system $S_{Ab} = \langle \mathcal{L}_{Ab},\ \vdash_{Ab},\ \mid\!\sim_{Ab},\ \Gamma_{Ab} \rangle$ that we describe in this chapter. This combination will correspond to our informal augmentation of $\mathcal{L}_{Prop}$, described back in Section 4.4, except that we can do this for arbitrary extended axiomatic formal systems with choice.

This chapter has three sections. Section 7.1 describes some constraints we put on S_1 so that the combination is well-defined. A description of S_{Ab} is provided in Section 7.2. In final Section 7.3, we show what the combination $S_{1,Ab} = S_{Ab} \cdot S_1$ looks like.

7.1 Constraints on S_1

We need to put one restriction on S_1 so that it successfully combines with S_{Ab} to have full retraction and full addition. This restriction is that the logical symbol $\perp \notin \Gamma_1$. Γ_1 can still be inconsistent, say by containing instances of P and $\neg P$, but it cannot contain $\perp$. Also, $\mathcal{L}_1$ and $\mathcal{L}_{Ab}$ must be disjoint, except possibly for non-logical boolean symbols. Call f the choice function associated with S_1.

Notice that this restriction on S_1 is purely syntactic, and does not interfere with or diminish the expressive content of S_1.

7.2 Properties of S_{Ab}

There is considerably much more to say about S_{Ab}. S_{Ab}'s sole purpose is to endow S_1 with full retractability and full addition. It can be thought of as a "scaffolding" put around $\langle \mathcal{L}_1, \vdash_1, \mid\!\sim_1, \Gamma_1 \rangle$ to give it full retraction and full addition, while preserving the inferential content of $\langle \mathcal{L}_1, \vdash_1, \mid\!\sim_1, \Gamma_1 \rangle$.

S_{Ab} is defined as follows:

1. $\mathcal{L}_{Ab} = Param \cup \{Ab, <\}$. $Param$ is a set consisting of countably infinitely many constant symbols, while Ab is a unary relation, and $<$ a binary one.

2. An $\mathcal{L}_{Ab}$-structure m_2 of $\mathcal{L}_{Ab}$ is a function such that $|m_2|$ is a non-empty set, and m_2 maps each symbol of $\mathcal{L}_{Ab}$ to its appropriate element:

   (a) For each $\ell \in Param$, $\ell^{m_2} \in |m_2|$

   (b) $<^{m_2} \subseteq |m_2|^2$

   (c) $Ab^{m_2} \subseteq |m_2|$

   This is just the standard definition of $\mathcal{L}$-structures.

3. The satisfaction relation $\models_{Ab}$ is the usual classical one for first-order languages.

4. $\mid\!\sim_{Ab}$ operates using a particular choice function g. We do not care about $\vdash_{Ab}$, as it will not be used in the combination. g selects the Ab-minimal models from a set W:

$$g(W) =_{def} \{x \in W \mid (\forall y \in W)[\neg y <_{Ab} x]\}, \tag{7.1}$$

   where W is a set of $\mathcal{L}_{Ab}$-structures. Proposition 5.3.1 shows that g satisfies coherence and contraction, so that so does g^* by Proposition 6.2.1.

   $<_{Ab}$ is the usual version of Ab-minimality:

$$
\begin{aligned}
m <_{Ab} n &\equiv_{def} |m| = |n| \wedge <^m = <^n \wedge (\forall \ell \in Param)[\ell^m = \ell^n] \wedge Ab^m \subset Ab^n \\
m \leq_{Ab} n &\equiv_{def} |m| = |n| \wedge <^m = <^n \wedge (\forall \ell \in Param)[\ell^m = \ell^n] \wedge Ab^m \subseteq Ab^n
\end{aligned}
\tag{7.2}
$$

5. Γ_{Ab} is the complete collection of all literals of the form $\pm\ell_i < \ell_j$ and $\pm\ell_i = \ell_j$, for each $\ell_i, \ell_j \in Param$. Γ_{Ab} formalizes $<$ as a total order on $Param$. It also asserts that the infinitely many constants in $Param$ are all distinct.

7.3 A Summary of $S_{1,Ab}$

Now that we have defined our $\mathcal{L}_{Ab}$, we can describe the combined system $S_{1,Ab}$ constructed according to Chapter 6, in all its gory detail:

$$
\begin{aligned}
Mod_{1,Ab}(\phi) &= \{(m_1, m_2) \in Struct(\mathcal{L}_1) \times Struct(\mathcal{L}_{Ab}) \mid (m_1, m_2) \models_{1,Ab} \phi\} \\
Th_{1,Ab}(W) &= \{\phi \in \mathcal{L}_{1,Ab} \mid (\forall (m_1, m_2) \in W)[(m_1, m_2) \models_{1,Ab} \phi]\} \\
\\
\mathcal{L}_{1,Ab} &= Closure(\mathcal{L}_1 \cup \mathcal{L}_{Ab}) \\
\Gamma \vdash_{1,Ab} \phi &\equiv f^*(Mod_{1,Ab}(\Gamma)) \subseteq Mod_{1,Ab}(\phi) \\
\Gamma \mathrel{|\!\!\sim}_{1,Ab} \phi &\equiv f^*(g^*(Mod_{1,Ab}(\Gamma))) \subseteq Mod_{1,Ab}(\phi) \\
f^*(g^*(W)) &= \{(m_1, m_2) \in g^*(W) \mid m_1 \in f([g^*(W)]_\ell)\} \\
g^*(W) &= \{(m_1, m_2) \in W \mid m_2 \in g([W]_r)\} \\
&= \{(m_1, m_2) \in W \mid (\forall (n_1, n_2) \in W)[\neg(n_2 <_{Ab} m_2)]\} \\
\Gamma_{1,Ab}(\Upsilon) &= \Upsilon \cup AB(\Gamma_1 \setminus \Upsilon) \cup \Gamma_{Ab}
\end{aligned}
\tag{7.3}
$$

We have labeled our semantic functions $Mod_{1,Ab}$ and $Th_{1,Ab}$ to stress the fact that they employ the satisfaction relation $\models_{1,Ab}$ over the class of $\mathcal{L}_{1,Ab}$-structures. Later on we leave this notation out to reduce clutter; it will be obvious from the context which satisfaction relation is being used. $\vdash_{1,Ab}$ ascertains truth in all f^* worlds, while $\mathrel{|\!\!\sim}_{1,Ab}$ looks at the f^*-preferred worlds that are *Ab*-minimal.

The one construction we have not yet explained is the last one in (7.3). The function AB *abnormalizes* its argument. If Γ is an $\mathcal{L}_1$-theory,

$$AB(\Gamma) =_{def} \{Ab(\sigma(\gamma)) \vee \gamma \mid \gamma \in \Gamma\} \tag{7.4}$$

What AB does is essentially make the formulas in Γ defeasible by attaching a disjunct $Ab(label)$ to each formula. σ is used to map each formula of $\mathcal{L}_1$ to some symbol of $Param$. [1] The construction in (7.3) allows us to make a distinction in $\Gamma_{1,Ab}$ between the hard and soft truths of Γ_1. Υ is the subset of Γ_1 meant to represent the hard truths, and are added as is to be a part of $\Gamma_{1,Ab}$. The remainder of Γ_1, $\Gamma_1 \setminus \Upsilon$ is considered the soft part, and is abnormalized by AB.

Our combined theory $\Gamma_{1,Ab}(\Upsilon) = \Upsilon \cup AB(\Gamma_1 \setminus \Upsilon) \cup \Gamma_{Ab}$, can thus be described as some set of "pure" $\mathcal{L}_1$ formulas from Γ_1, combined with the remaining formulas of Γ_1 which have been "labeled" with Abs, combined with the complete theory of $<$ and $=$ on $Param$ in Γ_{Ab}.

One final note: the assertion $Consis(\alpha, \Gamma)$ has a special interpretation in this context:

$$\begin{aligned} Consis(\alpha, \Gamma) \;\; &\equiv_{def} \;\; \Gamma \not\vdash_{1,Ab} \neg\alpha \\ &\iff \;\; f^*(Mod(\Gamma)) \not\subseteq Mod(\neg\alpha) \\ &\iff \;\; (\exists (m_1, m_2) \in f^*(Mod(\Gamma)))[(m_1, m_2) \models_{1,Ab} \alpha] \end{aligned} \tag{7.5}$$

The mission of Chapter 8 is to show that this combined system $S_{1,Ab}$ satisfies full retraction and full addition for formulas of $\mathcal{L}_1$. In other words, we can continually add sentences to the axioms of $S_{1,Ab}$ ($\Gamma_{1,Ab}(\Upsilon)$) to retract and add formulas of $\mathcal{L}_1$ regardless of whatever has already been added, of course provided these formulas do not directly conflict with any hard truths of $S_{1,Ab}$.

For the rest of this thesis, assume that $S_1 = \langle \mathcal{L}_1, \vdash_1, \mid\!\sim_1, \Gamma_1 \rangle$, $S_{Ab} = \langle \mathcal{L}_{Ab}, \vdash_{Ab}, \mid\!\sim_{Ab}, \Gamma_{Ab} \rangle$, and $S_{1,Ab} = \langle \mathcal{L}_{1,Ab}, \vdash_{1,Ab}, \mid\!\sim_{1,Ab}, \Gamma_{1,Ab} \rangle$ are as described above.

[1] For now, we treat σ as some arbitrary seeding function. Future research will explore various mappings σ and how they can help realize some further interesting structure within this framework. One example of some ideas on how to seed our formulas is explored in Section 11.4.

Chapter 8

$S_{1,Ab}$ Has Full Retraction and Addition

This chapter shows how the system $S_{1,Ab}$ described in Chapter 7 endows S_1 with full retraction and full addition for its formulas. The result is impressive, not only because we show how to implement additive elaboration tolerance for an arbitrary system S_1, but because the formulas which we have to add in order to retract and add other formulas are so simple. It is also reassuring to see that systems do exist with the properties of full retraction and addition, since the notions at first appear to be paradoxical.

We first explore the class of elaborative formulas $\Psi_{\mathcal{E}}$ used in our construction in Section 8.1. Section 8.2 then gives some key lemmas used to show that our interpretations of the Ab labels are flexible in various ways. We have to be careful that our choice functions (particularly g^*) do not return the empty set of models; Section 8.3 demonstrates how we manage to avoid these situations. We present the syntactic conditions under which full retractability works in Section 8.4, and then show full retraction and addition theorems for any $\mathcal{L}_1$-formulas in Section 8.5.

8.1 The Class of Elaborative Formulas $\Psi_{\mathcal{E}}$

An examination of the definitions of full retraction (Definition 5.5.1) and full addition (Definition 5.5.3) shows that whether a formula α is fully retractable and fully addable depends very sensitively on $\Psi_{\mathcal{E}}$, the class of elaborations we are allowed to apply to our representation. The reason is that we should be able to retract and add α by adding a particular formula, *no matter what intervening elaborations from $\Psi_{\mathcal{E}}$ have already been added to our representation.* This makes our proofs of full retraction and addition difficult, because we have to assume arbitrary formulas from $\Psi_{\mathcal{E}}$ may be a part of our representation. Thus, we want $\Psi_{\mathcal{E}}$ to be as simple and small as possible.

But on the other hand, $\Psi_{\mathcal{E}}$ has to be expressive enough to contain our elaborative formulas $\psi_{\emptyset}(\alpha)$ and $\psi_{+}(\alpha)$ for as many α as possible. Proving full retractability and addition boils down to solving a kind of fixed point equation for $\Psi_{\mathcal{E}}$.

The solution to this quandary is to "cut" this recursion by specifying a certain syntactic form for the formulas in $\Psi_{\mathcal{E}}$. It turns out that given this syntactic form, our theorems proceed without any problem. $Elab^{+}(\mathcal{L}_1, \mathcal{L}_{Ab})$ describes this special class of formulas:

Definition 8.1.1 ($Elab^{+}(\mathcal{L}_1, \mathcal{L}_{Ab})$). Let $\mathcal{L}_1$ and $\mathcal{L}_{Ab}$ be the languages as described in Chapter 7. $Elab^{+}(\mathcal{L}_1, \mathcal{L}_{Ab})$ is the set of all finite sentences of the form:

$$\begin{gathered} \phi \wedge \\ \bigwedge_i Ab(\ell_i) \vee \phi_i \wedge \\ \bigwedge_j (\forall_{Ab}\ x)[Ab(\ell_j) \wedge x < \ell_j \implies Ab(x)], \end{gathered} \tag{8.1}$$

where $\ell_i, \ell_j \in Param$, $\phi, \phi_i \in \mathcal{L}_1$, $\phi_i \neq \perp$, and $\forall_{Ab}$ refers to quantification over the domain of $\mathcal{L}_{Ab}$-structures.

We see that the elaborative formulas $\psi_{\emptyset}$ and ψ_{+} used in the simple system in Section 4.4 are represented in $Elab^{+}(\mathcal{L}_1, \mathcal{L}_{Ab})$. It is also easy to see that formulas in $Elab^{+}(\mathcal{L}_1, \mathcal{L}_{Ab})$ are closed under conjunction.

8.2 Retraction of Formulas and Labels

In this section we present two key lemmas that give us insight into the nature of the $\mathcal{L}_{1,Ab}$-models of $\Gamma_{1,Ab}$, even in the presence of any formula Ψ from $Elab^{+}(\mathcal{L}_1, \mathcal{L}_{Ab})$. The Upwardly Free *Ab* Lemma shows we can find $\mathcal{L}_{1,Ab}$-models of $\Psi \cup \Gamma_{1,Ab}$ where the extension of *Ab* is almost arbitrary. More specifically, if (m_1, m_2) is a model of $\Gamma_{1,Ab}$, we can arbitrarily tinker with m_2's extension of *Ab* over labels in $Param$ greater than those mentioned in $\Psi \cup \Gamma_{1,Ab}$, and still have a model of $\Psi \cup \Gamma_{1,Ab}$. In some sense, this should be obvious – if the labels from $Param$ are not mentioned in our axioms then clearly they cannot influence their truth, and are in some sense "free."

On the other hand, the Downwardly Free *Ab* Lemma shows we can also tinker a bit with the labels that *have been mentioned*. Given a model (m_1, m_2) of $\Gamma_{1,Ab}$, we can force *Ab* for additional elements of $Param$ less than or equal to those mentioned in $\Gamma_{1,Ab}$, and still have a model of $\Gamma_{1,Ab}$.

The Upwardly Free *Ab* Lemma is crucial for showing full retractability, as it shows that not only have we an infinite supply of unused labels from $Param$ to label our elaborations, but that these labels can later be consistently set to retract any of our statements, particularly those that might [logically] conflict with a later elaboration. The Downwardly Free *Ab* Lemma is important to the proof of full addition because it shows that we can have models even when our extension of labels mentioned are all contained in *Ab*.

Upwardly Free *Ab* Lemma. *Let Φ be any $\mathcal{L}_1$-theory and $\Psi \in Elab^{+}(\mathcal{L}_1, \mathcal{L}_{Ab})$. Say we have an $\mathcal{L}_{1,Ab}$-structure (m_1, m_2) such that $(m_1, m_2) \models_{1,Ab} \Phi \cup \Psi \cup \Gamma_{Ab}$, and call ℓ^{max} the $>$-highest symbol of $Param$ mentioned in Ψ.*

Call $\Lambda^{max} = \{\ell \in Param \mid \ell > \ell^{max} \in \Gamma_{Ab}\}$. Let Λ be a arbitrary subset of Λ^{max}. Then there is a model $(m_1, m_2^*) \models_{1,Ab} \Phi \cup \Psi \cup \Gamma_{Ab}$ where m_2^* looks just like m_2 except that $Ab^{m_2^*} = (Ab^{m_2} \setminus \Lambda^{max}) \cup \Lambda$.

Proof is in Section A.9.

In other words, we can assign the remaining parameters $>$-than those mentioned in Ψ arbitrarily to m_2's extension of *Ab* without affecting its satisfaction of $\Phi \cup \Psi \cup \Gamma_{Ab}$.

Downwardly Free *Ab* **Lemma.** *Let Φ be any $\mathcal{L}_1$-theory and $\Psi \in Elab^+(\mathcal{L}_1, \mathcal{L}_{Ab})$. Say we have an $\mathcal{L}_{1,Ab}$-structure (m_1, m_2) such that $(m_1, m_2) \models_{1,Ab} \Phi \cup \Psi \cup \Gamma_{Ab}$. Call ℓ^{max} the $>$-highest symbol of $Param$ mentioned in Ψ.*

Let n_2 be any $\mathcal{L}_{Ab}$-model of Γ_{Ab}, such that $\{a \in |n_2| \mid a \leq \ell^{max}\} \subseteq Ab^{n_2}$. (Note that this leaves n_2 unspecified for elements $>$-than ℓ^{max}.) Then $(m_1, n_2) \models_{1,Ab} \Phi \cup \Psi \cup \Gamma_{Ab}$.

Proof is in Section A.10.

8.3 Non-Empty Choice Functions

One problem with choice functions such as g^*/g, which picks out the models with minimal *Ab* extent, is that there could be no such minimal models, so that $g^*(W) = \emptyset$, even for a consistent set of formulas.[1]

A choice function g which returns the empty set on a non-empty input would wreck our construction. Given the complexity of our scaffolding it is not clear that g will tolerate this constraint, so we prove that it in fact does:

Lemma 8.3.1 [Every $<_{Ab}$ chain ends]. *Let $\Psi \in Elab^+(\mathcal{L}_1, \mathcal{L}_{Ab})$ and Φ an $\mathcal{L}_1$-theory. Say $(m_1^0, m_2^0) \models_{1,Ab} \Psi \cup \Phi \cup \Gamma_{Ab}$.*

Then there exists a $(n_1, n_2) \models_{1,Ab} \Psi \cup \Phi \cup \Gamma_{Ab}$ which is $\leq_{Ab}$-minimal amongst $Mod_{1,Ab}(\Psi \cup \Phi \cup \Gamma_{Ab})$ and $n_2 \leq_{Ab} m_2^0$.

Proof is in Section A.11

Corollary 8.3.1 [g^* has Minimal Elements]. *Say $(m_1^0, m_2^0) \models_{1,Ab} \Psi \cup \Phi \cup \Gamma_{Ab}$. Then there exists a $(n_1, n_2) \in g^*(Mod(\Psi \cup \Phi \cup \Gamma_{Ab}))$ such that $n_2 \leq_{Ab} m_2^0$.*

---

[1] [Etherington et al., 1985] gives an example for circumscription where the theory is consistent, but there are no P-minimal models:

$$\begin{array}{l}(\exists x)[P(x) \wedge (\forall y)[P(y) \implies x \neq s(y)]] \\ (\forall x)[P(x) \implies P(s(x))] \\ (\forall x, y)[s(x) = s(y) \implies x = y]\end{array} \tag{8.2}$$

Now, (8.2) clearly has a model, $\mathfrak{M}$, the most obvious one where $P^{\mathfrak{M}} = \{a, s^{\mathfrak{M}}(a), s^{\mathfrak{M}}(s^{\mathfrak{M}}(a)), \ldots\}$. A model $\mathfrak{M}'$ with a smaller extension of P would be $P^{\mathfrak{M}'} = \{s^{\mathfrak{M}'}(a), s^{\mathfrak{M}'}(s^{\mathfrak{M}'}(a)), \ldots\}$, and smaller still would be $P^{\mathfrak{M}''} = \{s^{\mathfrak{M}''}(s^{\mathfrak{M}''}(a)), \ldots\}$, and so forth. Since we have infinitely descending chains on *Ab*-minimal structures, we have no P-minimal models, resulting in an inconsistent circumscription.

This property is called smoothness in some circles. If an element is not minimal, it is less than some minimal element.

Proof. This is just a consequence of the definition of g^* and Lemma 8.3.1. $\square$

Corollary 8.3.2 [g^* **is Non-Empty**]. *Let* $\Psi \in Elab^+(\mathcal{L}_1, \mathcal{L}_{Ab})$ *and* Φ *a consistent* $\mathcal{L}_1$*-theory. Furthermore, if* Ψ *contains a conjunct with a pure* $\mathcal{L}_1$*-formula (*$\Psi \equiv \phi \wedge \bigwedge_i Ab(\ell_i) \vee \phi_i \wedge \bigwedge_j (\forall_{Ab}\ x)[Ab(\ell_j) \wedge x < \ell_j \implies Ab(x)]$*),* $\Phi \cup \phi$ *is consistent.*

Then $g^*(Mod(\Psi \cup \Phi \cup \Gamma_{Ab})) \neq \emptyset$.

Proof is in Section A.12.

8.4 Conditions for Full Retractability

With the help of the above lemmas, we can now give the syntactic conditions under which our $S_{1,Ab}$ is fully retractable, for any $\mathcal{L}_1$-formula α.

Full Retractability Lemma. *Let* $\Psi_{\mathcal{E}} = Elab^+(\mathcal{L}_1, \mathcal{L}_{Ab})$, τ *an element of* $Elab^+(\mathcal{L}_1, \mathcal{L}_{Ab})$, Φ *a consistent* $\mathcal{L}_1$*-theory, and* α *any* $\mathcal{L}_1$*-formula. Say* f*, the choice function underlying* S_1*, satisfies contraction, coherence, faithfulness, and* α*-reflection.*

Then $fully\text{-}retractable(\alpha, \langle \mathcal{L}_{1,Ab},\ \vdash_{1,Ab},\ \mid\!\sim_{1,Ab},\ \tau \cup \Phi \cup \Gamma_{Ab}\rangle)$.

Proof is in Section A.14.

8.5 Full Retraction and Addition Theorems

We finally can conclude that our construction $S_{1,Ab}$ was not in vain, that in fact we can fully retract and add formulas simply by adding the appropriate formula.

Full Retraction of $S_{1,Ab}$ **Theorem.** *Let* Γ_1 *be any* $\mathcal{L}_1$*-theory with* $\Upsilon \subseteq \Gamma_1$ *a consistent subset such that* $\Gamma_1 \setminus \Upsilon$ *is finite. Set* $\Psi_{\mathcal{E}}$ *be* $Elab^+(\mathcal{L}_1, \mathcal{L}_{Ab})$*. Let* f *be the choice function underlying* $\mid\!\sim_1$*.*

Say f *satisfies contraction, coherence, faithfulness and* α*-reflection for some* $\mathcal{L}_1$*-formula* α*.*

Then we have $fully\text{-}retractable(\alpha, \langle \mathcal{L}_{1,Ab},\ \vdash_{1,Ab},\ \mid\!\sim_{1,Ab},\ \Gamma_{1,Ab}(\Upsilon)\rangle)$.

Proof. By the Full Retractability Lemma, is enough to show that $\Gamma_{1,Ab}(\Upsilon)$ is of the same form as $\tau \cup \Phi \cup \Gamma_{Ab}$, where $\tau \in Elab^{+}(\mathcal{L}_1, \mathcal{L}_{Ab})$ and $\Phi \in \mathcal{L}_1$. Recall from (7.3), $\Gamma_{1,Ab}(\Upsilon) = \Upsilon \cup AB(\Gamma_1 \setminus \Upsilon) \cup \Gamma_{Ab}$. $\Upsilon \in \mathcal{L}_1$ and $AB(\Gamma_1 \setminus \Upsilon)$ being finite, is a member of $Elab^{+}(\mathcal{L}_1, \mathcal{L}_{Ab})$ so we are done. □

Full Addition of $S_{1,Ab}$ Theorem. *Let Γ_1 be any $\mathcal{L}_1$-theory with $\Upsilon \subseteq \Gamma_1$ a consistent subset such that $\Gamma_1 \setminus \Upsilon$ is finite. Set $\Psi_{\mathcal{E}}$ to be $Elab^{+}(\mathcal{L}_1, \mathcal{L}_{Ab})$. Let f be the choice function underlying $\mid\!\sim_1$.*

Say f satisfies contraction, coherence, faithfulness, and α-reflection for some $\mathcal{L}_1$-formula α. Then we have $fully\text{-}addable(\alpha, \langle \mathcal{L}_{1,Ab},\ \vdash_{1,Ab},\ \mid\!\sim_{1,Ab},\ \Gamma_{1,Ab}(\Upsilon)\rangle)$.

Proof is in Section A.15.

Chapter 9

Properties of $S_{1,Ab}$

Chapter 8 is heavily concerned with showing how and when an arbitrary extended axiomatic formal system with choice can be endowed with the properties of full retraction and full addition. We explore in this chapter other aspects of the system $S_{1,Ab}$ that we have constructed.

We first provide some additional motivating examples for how our construction works in Section 9.1. In Section 9.2 we provide various theorems showing that consistency is preserved from S_1 to $S_{1,Ab}$, and that the truths are preserved as well. Section 9.3 briefly explores the kinds of systems that can be augmented in this way; first-order logic is one of them.

We move on to more intriguing observations in Section 9.4. The syntax of the axioms in our original system S_1 actually influence what are the preferred worlds. This means that one can control the preference function $f^* \cdot g^*$, which operates on the semantic level, on the syntactic level. We provide some examples of how this is accomplished. The values of parameters can also be changed additively, provided the syntax of the theory follows some rules. An example is sketched out in Section 9.5. We show in Section 9.6 that our elaborated theory can be compacted to a natural form, which indicates that we may be able to model *epistemic entrenchments* in our formalisms in an elegant fashion. Finally, we conclude with a discussion of systems related to $S_{1,Ab}$ in Section 9.7.

9.1 Full Retraction and Addition in Action

Section 4.4 already gives some simple examples to show how our $\psi_\emptyset(\alpha)$ and $\psi_+(\alpha)$ work in propositional logic. This section gives some more examples when there are hard truths present.

Say $\mathcal{L}_1 = \{\alpha, \beta\}$, $\vdash_1 = \mid\!\sim_1 = \vdash$, the classical inference relation. $\Gamma_1 = \{\alpha, \alpha \implies \beta\}$. Say $Param = \mathbb{N}$ and $<$ the usual ordering on the numbers.

Example 9.1.1. Let us alter Example 4.4.1, this time letting $\alpha \implies \beta$ be a hard truth. Then our original set of axioms $\Gamma_{1,Ab}(\alpha \implies \beta)$ is

$$\begin{aligned} &Ab(1) \vee \alpha \\ &\alpha \implies \beta \end{aligned} \tag{9.1}$$

Again, in all Ab-minimal models, α and β hold. Now consider adding our $\psi_\emptyset = (Ab(3) \vee \beta) \wedge (Ab(4) \vee \neg\beta)$ again. This time, our Ab-minimal models can be divided into two classes:

$$\begin{aligned} &Ab(3) \wedge \alpha \wedge \beta \\ &Ab(1) \wedge Ab(2) \wedge \neg\alpha \wedge \neg\beta \end{aligned} \tag{9.2}$$

Again, we notice β has been retracted from our conclusions. Also, notice that in both of these classes the hard fact $\alpha \implies \beta$ still holds.

Example 9.1.2. We re-examine how Example 4.4.2 works in the presence of hard facts. For example, say that our original theory consisted of

$$\begin{aligned} &\alpha \\ &Ab(2) \vee \beta \\ &Ab(3) \vee (\alpha \wedge \beta \implies \gamma), \end{aligned} \tag{9.3}$$

where we are certain that α holds, but nothing else. Then when we add our $\psi_+(\neg\gamma)$ we end up with

$$\begin{aligned}
&\alpha \\
&Ab(2) \vee \beta \\
&Ab(3) \vee (\alpha \wedge \beta \implies \gamma) \\
&Ab(4) \vee \neg\gamma \\
&(\forall_{Ab}\ x)[Ab(4) \wedge x < 4 \implies Ab(x)]
\end{aligned} \tag{9.4}$$

which has Ab-minimal models:

$$\begin{aligned}
&Ab(2) \wedge \alpha \neg\beta \wedge \neg\gamma \\
&Ab(3) \wedge \alpha \ \ \wedge \beta \wedge \neg\gamma
\end{aligned} \tag{9.5}$$

In this case α is not allowed to be retracted, and we only have two possibilities, we retract β by asserting $Ab(2)$ or we retract $\alpha \wedge \beta \implies \gamma$ via $Ab(3)$.

9.2 Our Construction $S_{1,Ab}$ is Sound

We must show that our construction $S_{1,Ab}$ is sound – that is, any hard truths that held in S_1 before the conglomeration with S_{Ab} still hold in the new system. We show the same for soft facts. First we demonstrate a theorem about the consistency of $S_{1,Ab}$:

Theorem 9.2.1 [Satisfiability is Preserved]. *Let $S_1 = \langle \mathcal{L}_1,\ \vdash_1,\ \mid\!\sim_1,\ \Gamma_1 \rangle$ be an extended axiomatic formal system with choice. Say that $\Upsilon \subseteq \Gamma_1$ is satisfiable. Then the combined system $S_{1,Ab}$ has an $\mathcal{L}_{1,Ab}$-model.*

Proof is in Section A.16.

Note that Γ_1 could be unsatisfiable to begin with, but the transformation to $S_{1,Ab}$ will still provide models as long as the set of *hard* truths $\Upsilon \subseteq \Gamma_1$ is consistent.

Theorem 9.2.2 [Hard Truths are Preserved by Hard Consequence]. *Let α be an $\mathcal{L}_1$-formula and say $\Upsilon \subseteq \Gamma_1$ and $\Upsilon \vdash_1 \alpha$. Then $\Gamma_{1,Ab}(\Upsilon) \vdash_{1,Ab} \alpha$.*

Proof is in Section A.17.

Lemma 9.2.1 [g^*'s Behavior when Γ_1 is satisfiable.]. *Say Γ_1 is satisfiable. Then*

$$
\begin{aligned}
(m_1, m_2) \in g^*(Mod(\Gamma_{1,Ab}(\Upsilon))) \quad \Longleftrightarrow \quad & m_1 \models_1 \Gamma_1 \wedge \\
& Ab^{m_2} = \emptyset \wedge \\
& m_2 \models_2 \Gamma_{Ab}
\end{aligned}
\tag{9.6}
$$

Proof is in Section A.18.

Theorem 9.2.3 [Preservation of Soft Formulas]. *Let α be an $\mathcal{L}_1$-formula. Say Γ_1 is satisfiable and that $\Gamma_1 \mathrel{|\!\sim}_1 \alpha$. Then $\Gamma_{1,Ab}(\Upsilon) \mathrel{|\!\sim}_{1,Ab} \alpha$.*

Proof is in Section A.19.

9.3 Systems with Full Retraction and Addition

The Full Retraction Theorem and the Full Addition Theorem show that our construction $S_{1,Ab}$ will endow any extended axiomatic formal system with choice S_1 with full retraction and addition for an $\mathcal{L}_1$-formula α if it obeys the following properties:

1. f satisfies coherence, contraction, faithfulness and α-reflection.
2. $\Upsilon \subseteq \Gamma_1$ is consistent.
3. $\Gamma_1 \setminus \Upsilon$ is finite.

In this framework, this means any first order language $\mathcal{L}_1$ will be usable, and in fact, any first order language with $\mathrel{|\!\sim}_1$ defined classically (f is the identity) will be fully retractable/addable for *any* formula of $\mathcal{L}_1$!

Corollary 9.3.1 [Classical FOL Has Full Retraction and Addition]. *Let Γ_1 be a theory in some first-order language $\mathcal{L}_1$. Say $\Upsilon \subseteq \Gamma_1$ is some consistent subset such that $\Gamma_1 \setminus \Upsilon$ is finite. Then the system $\langle \mathcal{L}_{1,Ab},\ \vdash_{1,Ab},\ \mathrel{|\!\sim}_{1,Ab},\ \Gamma_{1,Ab}(\Upsilon)\rangle$ where $f = \mathbf{1}$ has full retraction and full addition, for any $\alpha \in \mathcal{L}_1$.*

Proof is in Section A.20.

In fact, we can generalize these results to any preferential semantics which are well-founded and satisfy α-reflection.

Corollary 9.3.2 [Preferential Semantics and Full Retraction and Addition]. *Let $\langle \mathcal{L}_1,\ \vdash_1,\ \mid\!\sim_1,\ \Gamma_1(\Upsilon)\rangle$ be an extended axiomatic formal system with choice function f. Say $f(X) =_{def} \{w \in X \mid (\forall x \in X)[x \leq w \implies x = w]\}$ for some well-founded relation $\leq$. Furthermore assume that f satisfies α-reflection.*

Say $\Upsilon \subseteq \Gamma_1$ is some consistent subset such that $\Gamma_1 \setminus \Upsilon$ is finite. Then the system $\langle \mathcal{L}_{1,Ab},\ \vdash_{1,Ab},\ \mid\!\sim_{1,Ab},\ \Gamma_{1,Ab}(\Upsilon)\rangle$ has full retraction and full addition for α.

Proof is in Section A.21.

9.4 The Power of Syntax in $S_{1,Ab}$

In Chapter 8, we showed how the combined system $\langle \mathcal{L}_{1,Ab},\ \vdash_{1,Ab},\ \mid\!\sim_{1,Ab},\ \Gamma_{1,Ab}(\Upsilon)\rangle$ can have full retraction and addition for all formulas in the original language $\mathcal{L}_1$. What we did not address is how to decide how to write the axioms of $\Gamma_{1,Ab}(\Upsilon)$. This is important, as the *syntax* of $\Gamma_{1,Ab}(\Upsilon)$, particularly, the structure of Γ_1 and how we assign labels to its formulas, can give rise to entirely different semantics in terms of $S_{1,Ab}$. This is a very important feature of our formalism, as it allows us to express preferences, usually accomplished only on the model-theoretic level, *within* our theory. We illustrate how syntax affects semantics in the next few examples:

Example 9.4.1 (Conjunctions). Let $\mathcal{L}_1 = \{\alpha, \beta\}$. Consider two theories $\Gamma_1 = \{\alpha, \beta\}$ versus $\Gamma_1' = \{\alpha \wedge \beta\}$. Let Υ be empty in both cases so that our "scaffolded" theories are $\Gamma_{1,Ab} = \{Ab(1) \vee \alpha, Ab(2) \vee \beta\}$ and $\Gamma_{1,Ab}' = \{Ab(1) \vee (\alpha \wedge \beta)\}$. Say we want to add the fact that $\neg\beta$ holds. Our elaborative formula for either case is $\psi_+(\neg\beta) = (Ab(3) \vee \neg\beta) \wedge (\forall_{Ab}\ x)[Ab(3) \implies x < 3 \implies Ab(x)]$. Adding this elaboration results in the two theories:

$$
\begin{aligned}
\psi_+(\neg\beta) \cup \Gamma_{1,Ab} \equiv \;& Ab(1) \vee \alpha \wedge \\
& Ab(2) \vee \beta \wedge \\
& Ab(3) \vee \neg\beta \wedge \\
& (\forall_{Ab}\, x)[Ab(3) \implies x < 3 \implies Ab(x)] \\
\psi_+(\neg\beta) \cup \Gamma'_{1,Ab} \equiv \;& Ab(1) \vee (\alpha \wedge \beta) \wedge \\
& Ab(3) \vee \neg\beta \wedge \\
& (\forall_{Ab}\, x)[Ab(3) \implies x < 3 \implies Ab(x)]
\end{aligned} \tag{9.7}
$$

We see that the preferred models of $\psi_+(\neg\beta) \cup \Gamma_{1,Ab}$ are $\{\alpha, \neg\beta\}$, while those of $\psi_+(\neg\beta) \cup \Gamma'_{1,Ab}$ include both $\{\alpha, \neg\beta\}$ and $\{\neg\alpha, \neg\beta\}$ – while α was independent of adding the elaboration that $\neg\beta$ in $\Gamma_{1,Ab}$, it was also retracted as a side-effect in $\Gamma'_{1,Ab}$.

Syntactically, this is obvious because in $\Gamma_{1,Ab}$ we assign each formula its own label, while in $\Gamma'_{1,Ab}$ we force α and β to share the same label. Intuitively, each *Ab* label *names* a particular formula, so that when the label is retracted, the *entire* associated formula must be retracted.

We will have more to say about how the *Ab* labels *partially reify* our formulas, and act as contexts on them [McIlraith, 2003] later in Section 11.4. Let us look at another example first:

Example 9.4.2 (Lightning and Thunder). This example is taken from [Subramanian and Genesereth, 1987]. Let $\mathcal{L}_1 = \{\ell, t\}$, where ℓ means "lightning is observed," while t is "thunder is observed."

Consider two separate, but equivalent theories, $\Gamma_1 = \{\ell, t\}$ versus $\Gamma'_1 = \{\ell, \ell \implies t\}$. Γ_1 represents two independent observations about lightning and thunder. Γ'_1 on the other hand, contains the fact that lightning was observed, but instead expresses the fact that lightning causes thunder. These theories are semantically equivalent, although they represent some different intuitions. It turns out our abnormalization operation is sensitive to this difference:

Say $\Gamma_{1,Ab} = \{Ab(1) \vee \ell, Ab(2) \vee t\}$, while $\Gamma'_{1,Ab} = \{Ab(1) \vee \ell, Ab(2) \vee (\ell \implies t)\}$. Now say we retract the fact that we observed lightning. This corresponds to the elaborative formula $\psi_\emptyset(\ell) = (Ab(3) \vee \ell) \wedge (Ab(4) \vee \neg\ell)$. Hence our updated theories are:

$$\begin{aligned} \psi_\emptyset(\ell) \cup \Gamma_{1,Ab} \;\equiv\; & Ab(1) \vee \ell \;\wedge \\ & Ab(2) \vee t \;\wedge \\ & Ab(3) \vee \ell \;\wedge \\ & Ab(4) \vee \neg\ell \\ \\ \psi_\emptyset(\ell) \cup \Gamma'_{1,Ab} \;\equiv\; & Ab(1) \vee \ell \;\wedge \\ & Ab(2) \vee (\ell \implies t) \;\wedge \\ & Ab(3) \vee \ell \;\wedge \\ & Ab(4) \vee \neg\ell \end{aligned} \tag{9.8}$$

In the first case, the preferred models of $\psi_\emptyset(\ell) \cup \Gamma_{1,Ab}$ are $\{\ell, t\}$ and $\{\neg\ell, t\}$ – t is independent of ℓ, since they are written to be separate observations. For the more causal $\psi_\emptyset(\ell) \cup \Gamma'_{1,Ab}$ we get three models: $\{\ell, t\}$, $\{\neg\ell, t\}$, and $\{\neg\ell, \neg t\}$ – in this causal theory, after retracting lightning, we are not even sure that thunder holds anymore.

This exercise shows that *how* we frame our theories is important, as it will affect how other facts may change in the presence of elaborations.

9.5 Changing Parameters

We can change the values of parameters in S_{Ab}, simply by adding the proper formula. Example 9.5.1 shows how this works, as well as how the syntax of the original theory is important. In particular, for this kind of elaboration to be successful, we must follow the design principles alluded to in Section 3.4.

Example 9.5.1 (Parameters). Adapted from [Parmar, 2002]: Assume $\mid\!\sim$ has an associated choice function f, and obeys inclusion and right conjunction. Let M be some parameter and m some value it can take. Say we have some set of axioms $\Gamma[M]$ with various occurrences of the formula/term $\alpha[M]$. Contrast this to the "instantiated" version $\Delta[M]$, which looks just like $\Gamma[M]$ but contains $\alpha[m]$ instead of $\alpha[M]$.

$$\Gamma[M] \mathrel{|\!\sim} M = m \quad \Delta[M] \mathrel{|\!\sim} M = m, \tag{9.9}$$

We will show that these theories act differently in the presence of a parameter change. Say we have some $n \neq m$ and that both theories have full addition. Further assume that $Consis(M = n, \Gamma[M])$ as well as $Consis(M = n, \Delta[M])$. By full addition, we know we have $\psi_+(M = n, \Gamma[M])$ and $\psi_+(M = n, \Delta[M])$ such that:

$$\begin{aligned} \psi_+(M = n, \Gamma[M]) \cup \Gamma[M] \mathrel{|\!\sim} M = n \\ \psi_+(M = n, \Delta[M]) \cup \Delta[M] \mathrel{|\!\sim} M = n \end{aligned} \tag{9.10}$$

Now say $\gamma[\alpha[M]]$ (respectively $\gamma[\alpha[m]]$) is some member of $\Gamma[M]$ ($\Delta[M]$). Since $\mathrel{|\!\sim}$ obeys inclusion we know that

$$\begin{aligned} \psi_+(M = n, \Gamma[M]) \cup \Gamma[M] \mathrel{|\!\sim} \gamma[\alpha[M]] \\ \psi_+(M = n, \Delta[M]) \cup \Delta[M] \mathrel{|\!\sim} \gamma[\alpha[m]] \end{aligned} \tag{9.11}$$

By right conjunction, then:

$$\begin{aligned} \psi_+(M = n, \Gamma[M]) \cup \Gamma[M] \mathrel{|\!\sim} M = n \wedge \gamma[\alpha[M]] \\ \psi_+(M = n, \Delta[M]) \cup \Delta[M] \mathrel{|\!\sim} M = n \wedge \gamma[\alpha[m]] \end{aligned} \tag{9.12}$$

Hence $\psi_+(M = n, \Gamma[M]) \cup \Gamma[M] \mathrel{|\!\sim} \gamma[\alpha[n]]$, but not necessarily so for $\psi_+(M = n, \Delta[M]) \cup \Delta[M]$. In fact, it is impossible to change $\gamma[\alpha[m]]$ by accessing M, as by inclusion, $\beta \cup \Gamma[M] \mathrel{|\!\sim} \gamma[\alpha[m]]$, for any β.

Example 9.5.1 is trivial in one way – of course $\gamma[\alpha[m]]$ will not get updated with the new value of $M = n$ since it does not depend on M. But this example showcases how important parameters are, and how additive elaborations can perform parameter changes, *as long as the parameter in question is "seeded" properly.* To change all occurrences of M from m to n in Γ, it was enough to add $\psi_+(M = n, \Gamma[M])$. The instantiated theory $\Delta[M]$ on the other hand, will require extensive brain surgery in order to replace all occurrences of $\alpha[n]$ with $\alpha[m]$. The problem will be compounded if there are other unrelated occurrences of m that are not supposed to be changed. This example highlights the importance of using the *unique roles assumption* as a guide in determining how to use parameters.

9.6 Compaction of $S_{1,Ab}$

As we add our elaborations from $Elab^+(\mathcal{L}_1, \mathcal{L}_{Ab})$, the set of axioms in our system $\langle \mathcal{L}, \vdash, \mid\!\sim, \Gamma \rangle$ grows and grows. Computationally (as well as conceptually), our set of axioms will become more and more complicated. Can we "compact" our theory to a simpler one which will admit the same elaborations?

It turns out the answer is yes, and the analysis of this compaction property of $S_{1,Ab}$ provides some further insights into the workings of $S_{1,Ab}$.

Throughout this section, assume $\Gamma_{1,Ab}(\Upsilon)$ to be as described in (7.3). Specifically, say it contains statements of the form $\phi \wedge \bigwedge_i Ab(\ell_i) \vee \phi_i$, where $\phi, \phi_i \in \mathcal{L}_1$, and $\ell_i \in Param$, plus statements Γ_{Ab} about the ordering of the constants ℓ in $Param$. Let $\Psi \in Elab^+(\mathcal{L}_1, \mathcal{L}_{Ab})$ consist of some sequence of elaborations $\Psi = \psi_1 \wedge \ldots \wedge \psi_n$, where each ψ_j is one of the four forms:

1. $\phi_j \in \mathcal{L}_1$
2. $Ab(\ell_j^+) \vee \phi_j$, where $\phi_j \in \mathcal{L}_1$,
3. $(Ab(\ell_j^+) \vee \phi_j) \wedge (Ab(\ell_j^-) \vee \neg\phi_j)$, where $\phi_j \in \mathcal{L}_1$, and
4. $(Ab(\ell_k) \vee \phi_k) \wedge (\forall_{Ab}\ x)[Ab(\ell_k) \wedge x < \ell_k \implies Ab(x)]$, where $\phi_k \in \mathcal{L}_1$.

Each of these four kinds of formulas corresponds to a kind of elaboration. The first is a hard truth, while the second, a soft one. The third is our retraction formula $\psi_\emptyset(\phi_j)$ and the fourth, our addition formula $\psi_+(\phi_k)$.

The theory $\Sigma = \Psi \cup \Gamma_{1,Ab}(\Upsilon)$ represents the trajectory of any abnormalized $\mathcal{L}_1$ theory as we tack on more and more elaborations. We want to show that we can always reformulate Σ to an equivalent, smaller set of formulas that will emulate its behavior, even under subsequent elaborations.

Briefly, $Compaction_1(\Sigma)$ rewrites the elaboration of the form $(\forall_{Ab}\ x)[Ab(\ell_k) \wedge x < \ell_k \implies Ab(x)]$ to $Ab(\ell_k) \implies Ab(\ell)$, for each $\ell < \ell_k$. This way of compacting theories is interesting because it detaches the theory's dependence on $<$, and explicitly models it using the implications. This points to a more general way of representing additive elaborations, where the epistemic entrenchment ordering is itself amenable to elaboration.

Definition 9.6.1 (*Compaction*$_1(\Sigma)$). Let Σ be of the form above. We can generally represent Σ as:

$$\begin{aligned}\Sigma \;=\; & \phi \cup \\ & \{Ab(\ell_j) \vee \phi_j \mid j \in J\} \cup \\ & \{(\forall_{Ab}\, x)[Ab(\ell_k) \wedge x < \ell_k \implies Ab(x)] \mid k \in K\} \cup \\ & \Gamma_{Ab},\end{aligned} \tag{9.13}$$

where $K \subseteq J$. Assuming Σ has the form in (9.13), we define the *Compaction*$_1$ of Σ to be:

$$\begin{aligned}Compaction_1(\Sigma) \;=_{def}\; & \phi \cup \\ & \{Ab(\ell_j) \vee \phi_j \mid j \in J\} \cup \\ & \{Ab(\ell_k) \implies Ab(\ell) \mid k \in K \wedge \ell \in \Sigma \wedge \ell < \ell_k \in \Gamma_{Ab}\} \cup \\ & \Gamma_{Ab}\end{aligned} \tag{9.14}$$

Essentially, *Compaction*$_1(\Sigma)$ removes all occurrences of the form $(\forall_{Ab}\, x)[Ab(\ell_k) \wedge x < \ell_k \implies Ab(x)]$ and replaces them with the consequences $Ab(\ell_k) \implies Ab(\ell)$, for each ℓ mentioned in Σ less than ℓ_k. This reformulation of Σ shows a more general way of characterizing our abnormalization of Γ_1. We see that the priority of statements is controlled not really by $<$, but by the relationship $Ab(\ell_k) \implies Ab(\ell)$ that is generated from the total order $<$. This detachment indicates there may be a richer characterization of dependence amongst formulas that will avoid the problem showcased in Example 4.4.3 of having to retract too much.

Rules of the form $Ab(\ell_k) \implies Ab(\ell_j)$ can be interpreted as epistemic entrenchments, as briefly mentioned at the end of Section 11.4. In terms of our framework, if our theory consists of

$$\begin{aligned} &Ab(\ell_k) \implies Ab(\ell_j) \\ &Ab(\ell_k) \vee \phi_k \\ &Ab(\ell_j) \vee \phi_j, \end{aligned} \tag{9.15}$$

then we will prefer having ϕ_k true over having ϕ_j be true, as if ϕ_k were false, both $Ab(\ell_k)$ and $Ab(\ell_j)$ would hold. In the other case, at most $Ab(\ell_j)$ would have to be true.

We provide a theorem showing that $Compaction_1(\Sigma)$ is equivalent to Σ, even under elaborations, for $\mid\!\sim_{1,Ab}$.

Theorem 9.7.1 [*$Compaction_1$* **equivalent to Σ under elaborations**]. *Let Σ be of the form above. Then for any $\Psi' \in Elab^+(\mathcal{L}_1, \mathcal{L}_{Ab})$ and $\phi \in \mathcal{L}_1$, we have:*

$$\Psi' \cup \Sigma \mid\!\sim_{1,Ab} \phi \iff \Psi' \cup Compaction_1(\Sigma) \mid\!\sim_{1,Ab} \phi \tag{9.16}$$

Proof is in Section A.23.

9.7 Related Systems to and Notes about $S_{1,Ab}$

[Winslett, 1989] has a system much like ours, which uses circumscription to compute minimal change. It also has a notion of *protected formulas* which are akin to our hard truths. Winslett uses a prioritization over predicates in the language. We argue that our system is epistemologically better than hers because we prioritize formulas, not predicates, which is more meaningful. Furthermore, we perform this prioritization within the language.

In their critique of approaches to belief revision, [Friedman and Halpern, 1996] notes that

> It is far from obvious that in a given epistemic state K we should allow arbitrary consistent formulas to be accepted. Intuitively, this does not allow for the possibility that some beliefs are held so firmly that their negations could never be accepted.

In our framework, the allowance for hard facts (via Υ) allows for the non-admittance for the contradiction of these hard facts in our future elaborations.

We have used the idea of abnormalizing our given theory in order to endow it with elaboration tolerance. This idea was first introduced in [Amir, 2000]. Amir shows under his metric that an abnormalized theory is more elaboration tolerant than the theory itself. This is not the case for general nonmonotonic theories and their monotonic equivalents however. (See Section 2.4 for more details.)

As mentioned in 4.4, [Boutilier, 1996] is similar to our formalism, in that if we add $\psi_+(\alpha)$ and then some more elaborations, and then $\psi_+(\neg\alpha)$, then all soft facts added before $\psi_+(\alpha)$ will be retracted. The difference between our formalism and Boutilier's is that in ours, *all* soft formulas are retracted up to $\psi_+(\alpha)$, while essentially only those asserted between $\psi_+(\alpha)$ and $\psi_+(\neg\alpha)$ are retracted in [Boutilier, 1996].

The mechanism that we have implemented looks also very similar to *truth maintenance systems* (TMSs), which explicitly represent dependencies between formulas so as to efficiently and consistently enable retractions and additions of formulas. There are two kinds of TMSs, justification-based (JTMS), and assumption-based (ATMSs) [de Kleer, 1986]. JTMSs record whether each datum is *in* (believed) or *out* (not believed). ATMSs record more information: each datum is instead labeled with the assumptions that support it being true. We can interpret our system S_{Ab} in terms of an ATMS, regarding our *Ab*s as labels for assumptions. Hence for each formula $\neg Ab(\ell) \implies \phi \in S_{Ab}$, $\neg Ab(\ell)$ implicitly represents the assumptions that allow ϕ to hold. The main difference between S_{Ab} and the ATMS is that the *Ab*-labels never explicitly reference this set of assumptions; rather they are inherently represented by the interactions of the *Ab*s through the precedence ordering of our labels and through the truths of the other $\mathcal{L}_1$-formulas to which they are attached. This existential quality of the *Ab*s with respect to assumptions is fundamental to the elaboration tolerance of S_{Ab} – since the underlying assumptions are never explicitly equated with a particular *Ab*, they are easier to change later on.

[Friedman, 1978] also uses a construction similar to abnormalization. This construction is known as *Friedman's translation*, where one adds a disjunct to a formula. This construction is used to prove that Heyting arithmetic (HA – the intuitionistic version of Peano arithmetic (PA)) is closed under *Markov's rule*. This result allows one to show that "every Turing machine program that, provably in PA, converges at all arguments, also does provably in HA." The same translation is used to prove a similar result, that the provably recursive functions and provable ordinals of classical and intuitionistic set theory are the same. In both cases, Friedman's translation fosters a clean navigation of the strict constructive proof rules of HA.

The relationship between intuitionistic systems, which only derive a subset of the conclusions of the classical ones, and elaboration tolerance is a task for future work. The extra formulas appended by Friedman's translations provide "outs," or places where we can insert more information into our rules as needed (similar to the hooks used in Emacs functions). It could be that the rules of intuitionistic logic, which are constructive by nature, work synergistically in the presence of these outs. [1]

[van Eijck and Kamp, 1997] presents the notion of *discourse representation theory*, meant to describe the semantics of multi-sentence discourse. It is based on the view that

> each new sentence S of a discourse is interpreted in the context provided by the sentences preceding it. ...one and the same structure serves simultaneously as content and as context.

Discourse representation theory is mainly used to handle anaphora between sentences. This approach shares our view that discourse is a dynamic object, perpetually prone to update by future utterances. But the two systems have different focuses. Our system S_{Ab} focuses on the retraction and addition of formulas; the basic entities are formulas. The one in [van Eijck and Kamp, 1997] on the other hand is primed to represent anaphora, which are relationships between objects mentioned in sentences. S_{Ab} does not work at a sufficient level of detail to deal with these structures. It is possible, however, that a more general construction, explored in Section 11.4, can represent this capability.

[1] Thanks to Professor Grigori Mints for the reference.

Part III

Extensions

Chapter 10

Elaboration Tolerance and AI Formalisms

In this chapter we explore some relationships between other fields of AI and elaboration tolerance. Belief revision (Section 10.1) and reformulation (Section 10.2) have many similar characteristics. In fact some of the tenets of belief revision (the AGM postulates) can be adapted to provide additional intuitive constraints on $S_{1,Ab}$. But overall, elaboration tolerance, particular our implementation of additive elaboration tolerance, differs from the two fields in important ways.

This thesis has mainly focused on the elaboration tolerance of logical structures. To balance this out, we also explore how elaboration tolerance relates to Bayesian networks and neural networks in Sections 10.3 and 10.4.

10.1 Belief Revision

Elaboration tolerance looks very similar to belief revision as they both involve change to a knowledge base. In this chapter we examine how these areas of research overlap, and how they differ.

10.1.1 Overview of Belief Revision

Belief revision [Gärdenfors, 1992] is concerned with how one revises one's beliefs, given some specified change. Most work on belief revision uses sets of formulas (usually propositions, but sometimes first-order logic) to represent their knowledge base K. In this framework, updates to K are implemented by means of set-theoretical operations combined with some logical operators. There are three different kinds of belief changes:

1. expansion,
2. revision, and
3. contraction.

In *expansion*, one simply adds some formula ψ to the set of formulas in K. This operation is simple set union, which means it could very well create an inconsistent set of formulas. *Revision* on the other hand, is smarter, in that it changes other facts in order to accommodate the expected revision, preserving consistency at all costs. So for example, if α and $\alpha \implies \beta$ are in the current set of beliefs, and we want to revise our beliefs to include $\neg\beta$, the revision operator will either retract α or $\alpha \implies \beta$, or both. Finally *contraction* removes some formula from the knowledge base K. It is like revision, in that not only is the desired formula ψ removed, but surgery is performed on K to ensure ψ is not concluded by other means.

These sets of formulas, called *belief states*, are usually closed under logical consequence. Some researchers disapprove of this notion, as this logical closure tends to lose the information about what formulas were original and which were derived. Instead *bases of belief sets* are used, which are subsets of the belief set, to distinguish the original beliefs from those which are derived by inference.

Other researchers use possible worlds to model belief sets. The intuition here is that people, when describing their beliefs, do not keep track of some set of sentences representing their beliefs, but imagine truth in terms of possible worlds. In general, belief sets are preferred for computation, as they are a more tractable paradigm than possible worlds.

Foundations versus Coherence Approach

There are two different philosophies describing how people change their beliefs, the *foundations* versus the *coherence* approach. In the foundations approach, also known as the *justifications* approach, every reason for a belief is included in a model. One holds a belief to be true, as long as there is a reason to do so. If there is no reason to have a belief, then the belief is removed from the belief set. In this paradigm, the semantics of deciding what to believe given an update is completely decided by the structure of the justifications.

In the coherence approach, the pedigrees of a belief are *not* preserved. This approach instead focuses on the logical relations between formulas, and reacts to change by preserving consistency while minimizing change. The famous AGM postulates [Alchourrón et al., 1985] (described in detail later) is an example of the coherence approach. These are rules which govern contraction and revision operations, specifying which logical relationships they should satisfy. They work on sets of logical sentences, and try to preserve some kind of logical harmony between beliefs.

Epistemic Entrenchment

One important feature of belief states is *epistemic entrenchment*, which measures how strongly one holds a particular belief. This measure is used to govern revisions and contractions – if one has to choose between two different beliefs to abandon, one can drop the one which is less epistemically entrenched. As discussed earlier, as well as later in Section 10.1.2, we do implement a form of epistemic entrenchment via our Abs in $S_{1,Ab}$. Our approach is slightly more expressive in that the *truth* of our formulas affects how they interact. Also most entrenchment orderings are static, while ours is changeable over time.

The AGM Postulates

A tour of belief revision would be incomplete without mention of the *AGM postulates*. The AGM postulates [Alchourrón et al., 1985] characterize the desirable properties one would like a belief revision and belief contraction operator to have. There are eight for revision and eight for contraction, and there are representation theorems that show that if a revision/contraction operator obeys various subsets of these postulates, the operator in question can be defined in terms of a *selection function*.

Let K represent a belief set (a logically closed set of formulas), and ϕ some other logical formula. The notation $K \dot{-} \phi$ is the belief state resulting in *contracting* ϕ from K, that is, removing it from K. $K \dot{+} \phi$ on the other hand is the belief state resulting from *revising* ϕ in K. The notation $K + \phi$ is simply the result of adding the formula ϕ to K, and closing the result under standard consequence. Note that $K_\perp$ refers to the inconsistent belief state, which consists of all formulas in the language.

The AGM postulates for contraction are [Gärdenfors, 1992]:

(1c) For any sentence ϕ and belief set K, $K \dot{-} \phi$ is a belief set.

(2c) $K \dot{-} \phi \subseteq K$.

(3c) If $\phi \notin K$, then $K \dot{-} \phi = K$

(4c) If not $\vdash \phi$, then $\phi \notin K \dot{-} \phi$

(5c) If $\phi \in K$ then $K \subseteq (K \dot{-} \phi) + \phi$

(6c) If $\vdash \phi \iff \psi$ then $K \dot{-} \phi = K \dot{-} \psi$

(7c) $(K \dot{-} \phi) \cap (K \dot{-} \psi) \subseteq K \dot{-} (\phi \wedge \psi)$

(8c) If $\phi \notin K \dot{-} (\phi \wedge \psi)$ then $K \dot{-} (\phi \wedge \psi) \subseteq K \dot{-} \psi$.

The AGM postulates for revision are:

(1r) For any sentence ϕ and belief set K, $K \dot{+} \phi$ is a belief set.

(2r) $\phi \in K \dot{+} \phi$.

(3r) $K \dot{+} \phi \subseteq K + \phi$.

(4r) If $\neg\phi \notin K$ then $K + \phi \subseteq K \dot{+} \phi$.

(5r) $K \dot{+} \phi = K_\perp$ iff $\vdash \neg\phi$.

(6r) If $\vdash \phi \iff \psi$ then $K \dot{+} \phi = K \dot{+} \psi$.

(7r) $K \dot{+} (\phi \wedge \psi) \subseteq (K \dot{+} \phi) + \psi$.

(8r) If $\neg\psi \notin K \dot{+} \phi$, then $(K \dot{+} \phi) + \psi \subseteq K \dot{+} (\phi \wedge \psi)$.

These sets of axioms are parallel. Items (1c) and (1r) assert that the contractions and revisions are closed under logical consequence (the definition of a belief set). Rules (2c), (3c), (4c), and (2r), (3r), (4r), and (5r) restrict the altered belief sets compared to the original belief set, particularly when the revision/contraction is somewhat trivial, such as for contraction, when ϕ does not need to be retracted, or in revision, when it is consistent to add ϕ. (5c) is a *recovery postulate* which asserts how the effect of undoing a contraction is related to the original set. (6c) and (6r) simply assert that the contractions and revisions respect logical equivalence, while (7c), (8c), (7r) and (8r) show how the contractions and revisions relate to the conjunction operator $\wedge$.

There are two identities which define the revision operator in terms of the contraction one, and vice versa:

$$\begin{array}{lll} K \dot{+}_{LI} \phi & =_{def} & (K \dot{-}_{LI} \neg\phi) + \phi \qquad\qquad \text{Levi Identity} \\ K \dot{-}_{HI} \phi & =_{def} & K \cap (K \dot{+}_{HI} \neg\phi) \qquad\qquad \text{Harper Identity} \end{array} \tag{10.1}$$

If $\dot{-}_{LI}$ satisfies all the contraction postulates except (5c), then $\dot{+}_{LI}$ defined by the Levi Identity satisfies all eight revision postulates. Also, if $\dot{+}_{HI}$ satisfies all eight revision postulates, then $\dot{-}_{HI}$ will satisfy all eight contraction ones.

Another interesting fact about the AGM postulates is that they characterize how $\dot{-}$ operates over worlds in terms of selection functions. Let $K \perp \phi$ be those maximal subsets of K, closed under logical consequence, that fail to imply ϕ, that is

$$\begin{aligned} K \perp \phi =_{def} \{U \subseteq K\}\{&U = Conseq(U) \wedge \phi \notin U \wedge \\ &(\forall V \subseteq K)[U \subset V \wedge V = Conseq(V) \implies \phi \in V]\}. \end{aligned} \tag{10.2}$$

Then a *selection function* γ is a function that chooses some subset of sets from $K \perp \phi$. Given this function, one can define contraction in terms of this *partial meet contraction function* as follows:

$$K \dot{-} \phi \quad =_{def} \quad \bigcap \gamma(K \perp \phi). \tag{10.3}$$

In other words, to contract ϕ from K, we take some subsets of $K \perp \phi$, take the intersection, and return that as our new belief set $K \dot{-} \phi$. What is fascinating is that there is a theorem indicating that $\dot{-}$ can be defined in terms of some partial meet contraction function γ *iff* it satisfies postulates (1c) – (6c). This is an intriguing result because it shows that the AGM postulates are enough to characterize a kind of background semantics.

Update versus Revision

Finally, we should distinguish between two types of change to a theory: *belief update* and *belief revision* [Katsuno and Mendelzon, 1992], first pointed out by [Keller and Wilkins, 1985]. *Belief revision* deals with properly describing a static world, while *belief updates* are used to properly describe that the world has changed. Update is more intrusive since it does not care as much about mutual consistency of facts, but that the model properly represents the updated world.

The two approaches have different semantics as well. To revise K with ϕ, any revision function satisfying the AGM postulates can be viewed *in terms of finding the models of ϕ that are closest to those of K*. Update on the other hand, is more complicated. For each model $\mathfrak{M}$ of K, we find the set of models of ϕ that are closest to $\mathfrak{M}$. The resulting theory is the one which holds in the union of these closest models, for each model $\mathfrak{M}$ of K. [Katsuno and Mendelzon, 1992] gives AGM style postulates for this kind of update, as well as a notion of "erasure." Erasing ϕ corresponds to representing a change to the world where ϕ may not be true.

We believe that elaborations perform both of these kinds of alterations. We often elaborate our representations in order to clear up any confusions, as in "there is no bridge across the river" with respect to the missionaries and cannibals problem, as well as to purposely change the listener's view of the world: "change the number of missionaries and cannibals from three to four." In our opinion, we do not need to distinguish between these two kinds of alterations, as we are more concerned with issues of representation, and both of these phenomena are indistinguishable on that level. In terms of semantics, we believe that both of these kinds of changes can be modeled by the behaviors of our choice function f under addition of formulas.

10.1.2 Elaboration Tolerance versus Belief Revision

Both elaboration tolerance and belief revision involve some kind of change to a knowledge base. While belief revision focuses on what beliefs should stay/be added after such a change, elaboration tolerance is more concerned with the properties of the underlying representational mechanism characterizing such change within the logic. In practice, the purpose of elaboration tolerance is the opposite of belief revision; while belief revision is concerned with the *effects* of adding or deleting a certain sentence, elaboration tolerance cares more about *what formula must be added or deleted to incur a certain change* [McCarthy, 1998].

Additive elaboration tolerance lies one level below belief revision, in that it focuses on the way change can be represented and implemented *compactly, within the representation.* That said, elaboration tolerance shares belief revision's concern in deciding what other facts should remain in the knowledge base. A theory of elaboration tolerance would be useless if it did not also properly characterize the "inertia" of unrelated statements after a change. For example, after changing the number of missionaries and cannibals from three to four, we should infer that there are a total of eight people participating in the scenario, but still believe there is only one boat.

Elaboration tolerance and belief revision interpret their changes in subtly different ways. When applying a change to the set of beliefs, belief revision (particularly in the coherence approach) uses two main criteria to accommodate the change: the logical relationship between formulas, and the epistemic entrenchment. Since any change must preserve consistency, formulas may need to be removed to accommodate a new added formula. Epistemic entrenchment is used to decide which formulas are to be removed. In elaboration tolerance, how to decide which formulas to retract, in addition to the one specified, depends entirely on the *meaning* of the formulas. While logical consequence is one aspect that is necessary, it only skims the surface.

There are all sorts of semantic entanglements that need to be represented within the theory, that go far beyond even epistemic entrenchment, although that can be used as a means to model some of the dependencies. Hence, elaboration tolerance requires a finer view of the meanings of the formulas, and how they depend on each other. This is one of the main reasons we have relied so heavily on a nonmonotonic consequence relation, which we have used as a means to enforce all these dependencies, without articulating them fully within the logic.

On another plane, characterizing inertia should be much simpler for elaboration tolerance than belief revision. Since elaboration tolerance is often based on common communication conventions, the form of inertia it employs should be unique, being used by all agents. Belief revision, on the other hand, must be able to model a multiplicity of inertias, since the kind of inertia could be agent-specific.

Our study of additive elaboration tolerance has been greatly motivated by human discourse, and to a large extent we have copied many aspects of discourse. For example, we distinguish between utterances, and the facts embedded in them. For example, $\psi_{\emptyset}(\alpha)$ means "forget α", while $\psi_{+}(\alpha)$ means to "assert α for now." These are very different from the sentences $\neg\alpha$ and α, which simply assert the falsity/truth of α. Elaboration tolerant structures generalize those structures of belief revision, because they represent, *within the logic*, the actions that are performed upon the database. This allows them to include the history of change that has taken place. This is important, because often an elaboration depends on the statements that have been previously uttered: "forget that last sentence." Belief structures which operate by way of sets of propositions manipulated by logical operators do not retain this information and hence cannot support these more expressive elaborations.

As mentioned before, there are two ways to view a representation. The first is as reified facts, each of which can be manipulated. The second is in terms of the models (possible worlds) that are described by the set of sentences. In some sense, we need both interpretations. The first is necessary because we often refer to distinct sentences in our discourse, as in "scratch that last sentence." But we also need possible worlds as a clean construct within which to describe other intuitions, such as minimal change and epistemic entrenchments. Our approach is flexible because it allows for both modalities. By labeling our sentences with the $Ab(\ell)$s we have somewhat reified our sentences, but we view our transformations in terms of the choice functions. As shown in Examples 9.4.1 and 9.4.2, our approach is sensitive to the syntax of our theories, while most belief revision approaches are not (particularly those that do not use bases of belief sets).

10.1.3 $S_{1,Ab}$ and Belief Revision

It is illuminating to see how our system $S_{1,Ab}$ for adding and retracting formulas fits in the framework of belief revision. For example, $S_{1,Ab}$, follows the spirit of the justifications approach, but encodes these justifications in such a way that we only care about the logical consistency of the resulting theory (corresponding to the coherence approach). The justifications for a belief are inversely represented by the form of our statements in $S_{1,Ab}$: $Ab(\ell) \vee \phi$ means "*unless* we have information to the contrary, assume that ϕ holds." Then we can unjustify ϕ by indirectly asserting $Ab(\ell)$.

Unlike the approaches to belief revision which employ belief sets, our $S_{1,Ab}$ can in fact distinguish between the original beliefs and those that are derived. This is because our labels $Ab(\ell)$ "reify" our sentences to the extent that they can be referenced, allowing them to be distinguished from their consequences. We can view our abnormalization operation AB as creating a *basis of a belief set* from our set of logical sentences by appending the disjuncts $Ab(\ell)$. We have further flexibility in being able to denote *hard facts* which are not subject to revision.

As mentioned at the end of Section 9.6, the encoding of the justifications leads to a simple sort of epistemic entrenchment. The justifications, or relations between our semantic content in $\mathcal{L}_1$, are related by formulas of the form $Ab(\ell_k) \implies Ab(\ell_j)$. If we have $Ab(\ell_k) \vee \phi_k$ and $Ab(\ell_j) \vee \phi_j$, and we have to assume one of these is false in order to properly revise our knowledge base, it is better to pick ϕ_j instead of ϕ_k, in order to minimize the extension of Ab. Hence $Ab(\ell_k) \implies Ab(\ell_j)$ means ϕ_k is more epistemically entrenched than ϕ_j. But since we can always add $Ab(\ell_j) \implies Ab(\ell_k)$ and cancel out this dependency, this epistemic entrenchment is itself modifiable.

These Ab labels provide a mild level of *illocutionary force* to our sentences, in that aspects *about* the message are represented as well, via the Ab labels. Most approaches in belief revision do not have this modality.

We find it satisfying that our S_{Ab} partly satisfies two major critiques of belief revision [Friedman and Halpern, 1996]:

1. how the epistemic state of an agent is modeled, and
2. the status of observations

The first bullet refers to the fact that an epistemic state may not be expressive enough. For example, we may also want to include beliefs about beliefs. By asserting our retraction and addition formulas $\psi_{\emptyset}(\alpha)$ and $\psi_{+}(\alpha)$ within the language itself, we do allow some of this modality. What is even more important is that we allow statements about the relative strengths of beliefs (in the compacted form $Ab(n) \implies Ab(m)$) *within* our language, which fulfills [Friedman and Halpern, 1996]'s requirement:

> the fact that an agent's epistemic state is characterized by a collection of formulas means that the epistemic state cannot include information about relative strength of beliefs (as required for the approach of, say, [Gärdenfors and Makinson, 1988]), unless this information is expressible in the language.

Another criticism of belief states is that there is no way to represent beliefs which are held so firmly that their negations could never be accepted. This relates to the second bullet, in that there is no distinction between facts versus beliefs. S_{Ab} does represent this difference through the hard and soft facts – what Friedman and Halpern call knowledge and belief.

10.1.4 Formal Connections between Frameworks

Section 10.1.1 refers to the representation theorem which states that any function $\dot{-}$ obeying the AGM postulates for contraction can be described in terms of a selection function γ. The contraction $K \dot{-} \phi$ is equivalent to the intersection of some subset (chosen by γ) of the maximal subsets of K which do not imply ϕ, as shown in (10.2). This selection function looks very much like our choice function. Although γ works in terms of sets of formulas, it can be viewed as choosing the set of possible worlds in which as many facts of K hold as possible, but where ϕ does not hold universally. Our choice function $f^* \cdot g^*$ on the other hand, simply chooses some subset of models of $\psi_\emptyset(\phi, \Gamma) \cup \Gamma$ which is not entirely contained in $Mod(\phi)$. γ is slightly different from $f^* \cdot g^*$ in that γ directly imbues semantics to a revision by ϕ, whereas $f^* \cdot g^*$'s primary purpose is to generate consequences, and only as a by-product provides the results of a revision. This difference in fundamental semantics will affect how we interpret the AGM postulates in terms of $S_{1,Ab}$.

$S_{1,Ab}$ and the AGM Postulates for Contraction

It is intriguing to examine how our retraction operation of adding $\psi_\emptyset(\alpha)$ to our sentences relates to the AGM postulates for contraction. From now on, we will interpret the notation $K \dot{-} \phi$ as $\{\phi \mid \psi_\emptyset(\alpha, \Gamma) \cup \Gamma \mid\!\sim_{1,Ab} \phi\}$, where $\Gamma = AB(K)$. The AGM postulates work in a paradigm where any truth is retractable. To make our formalism compatible, we assume that Γ is fully retractable for α.

Finally, we should note that our retraction operator has slightly different semantics from contraction. When we retract α, we are *definitely* ensuring that α is forgotten – neither α nor $\neg\alpha$ can be inferred. Standard contraction on the other hand is one-sided, only caring about removing α as a consequence. If $\neg\alpha$ held, the result of the contraction would be unchanged. However, we can reinterpret the AGM rules, while keeping within the spirit of the postulates, to find interesting correspondences and constraints of AGM with $S_{1,Ab}$. That is, some of the AGM postulates serve to constrain our choice functions f and g further, while others are verified by our paradigm.

We provide a sketch of these relations next.

1. For any sentence ϕ and belief set K, $K \dot{-} \phi$ is a belief set.

   We define a belief set as just a set of sentences closed under $\mid\!\sim_{1,Ab}$, so this property is true by definition.

2. $K \dot{-} \phi \subseteq K$.

   This corresponds to the requirement that

$$[f^*(g^*(Mod(\Gamma)))]_\ell \subseteq [f^*(g^*(Mod(\psi_\emptyset(\phi,\Gamma) \cup \Gamma)))]_\ell, \tag{10.4}$$

   which means that as we change our models to include models where ϕ does not hold, we must keep all of the models from before.

   To show (10.4), we must prove that for any $(a_1, a_2) \in f^*(g^*(Mod(\Gamma)))$, then there is some a_2^* such that $(a_1, a_2^*) \in f^*(g^*(Mod(\psi_\emptyset(\phi,\Gamma) \cup \Gamma)))$, or that

$$\begin{aligned} (a_1, a_2) \in g^*(Mod(\Gamma)) \wedge a_1 \in f([g^*(Mod(\Gamma))]_\ell) \implies \\ (\exists a_2^*)[(a_1, a_2^*) \in g^*(Mod(\psi_\emptyset(\phi,\Gamma) \cup \Gamma)) \wedge \\ a_1 \in f([g^*(Mod(\psi_\emptyset(\phi,\Gamma) \cup \Gamma))]_\ell)] \end{aligned} \tag{10.5}$$

   By the Monotonic Retraction Lemma, $[g^*(Mod(\Gamma))]_\ell \subseteq [g^*(Mod(\psi_\emptyset \cup \Gamma))]_\ell$, so (10.5) reduces to

$$a_1 \in f([g^*(Mod(\Gamma))]_\ell) \implies a_1 \in f([g^*(Mod(\psi_\emptyset(\phi,\Gamma) \cup \Gamma))]_\ell)], \tag{10.6}$$

   which asserts that f must be monotonically increasing on addition of $\psi_\emptyset$, from $[g^*(Mod(\Gamma))]_\ell$ to $[g^*(Mod(\psi_\emptyset \cup \Gamma))]_\ell$. This will be trivially satisfied if f is monotonically increasing in general. Note that normal logical consequence, where f is the identity, satisfies this requirement.

3. If $\phi \notin K$, then $K \dot{-} \phi = K$

   This rule asserts that if K does not already entail ϕ, then the contraction does not change anything. The corresponding rule should be:

$$\begin{aligned} & f^*(g^*(Mod(\Gamma))) \not\subseteq Mod(\phi) \wedge f^*(g^*(Mod(\Gamma))) \not\subseteq Mod(\neg\phi) \implies \\ & \qquad [f^*(g^*(Mod(\psi_\emptyset(\phi,\Gamma) \cup \Gamma)))]_\ell = [f^*(g^*(Mod(\Gamma)))]_\ell. \end{aligned} \tag{10.7}$$

That is, if Γ neither entailed ϕ or $\neg\phi$ originally, adding $\psi_\emptyset(\phi)$ should not have changed any conclusions.

To show this, by the $[f^*]_\ell$-Equivalence Lemma, it is enough to show that $[g^*(Mod(\Gamma))]_\ell = [g^*(Mod(\psi_\emptyset(\phi,\Gamma) \cup \Gamma))]_\ell$, or by the Monotonic Retraction Lemma, that $[g^*(Mod(\Gamma))]_\ell \supseteq [g^*(Mod(\psi_\emptyset(\phi,\Gamma) \cup \Gamma))]_\ell$.

Since there are models of both ϕ and $\neg\phi$ in $g^*(Mod(\Gamma))$, it would seem that adding $\psi_\emptyset(\phi)$ to Γ should not change the $\mathcal{L}_1$ models, but this remains to be formally shown.

4. If not $\vdash \phi$, then $\phi \notin K \dot{-} \phi$

   The essence of this rule is that as long as ϕ is not a tautology, retracting it from Γ will be properly accomplished. This can be translated to a version of the definition of full retraction, proved to be correct in the Full Retraction Theorem.

$$\begin{aligned} & f^*(Mod(\Gamma)) \not\subseteq Mod(\phi) \wedge f^*(Mod(\Gamma)) \not\subseteq Mod(\neg\phi) \implies \\ & \qquad f^*(g^*(Mod(\psi_\emptyset(\phi,\Gamma) \cup \Gamma))) \not\subseteq Mod(\phi) \wedge \\ & \qquad f^*(g^*(Mod(\psi_\emptyset(\phi,\Gamma) \cup \Gamma))) \not\subseteq Mod(\neg\phi) \end{aligned} \tag{10.8}$$

5. If $\phi \in K$ then $K \subseteq (K \dot{-} \phi) + \phi$.

   This rule asserts that if ϕ is a consequence of Γ, then retracting it, and then adding it again as a hard fact will contain the same consequences as before the operations.

$$\begin{aligned} & f^*(g^*(Mod(\Gamma))) \subseteq Mod(\phi) \implies \\ & \qquad [f^*(g^*(Mod(\phi \cup \psi_\emptyset(\phi,\Gamma) \cup \Gamma)))]_\ell \subseteq [f^*(g^*(Mod(\Gamma)))]_\ell \end{aligned} \tag{10.9}$$

6. If $\vdash \phi \iff \psi$ then $K \dot{-} \phi = K \dot{-} \psi$

$$\vdash \phi \iff \psi \implies f^*(g^*(Mod(\psi_\emptyset(\phi, \Gamma) \cup \Gamma))) = f^*(g^*(Mod(\psi_\emptyset(\psi, \Gamma) \cup \Gamma))) \tag{10.10}$$

   If $\vdash \phi \iff \psi$ then $Mod(\psi_\emptyset(\phi, \Gamma) \cup \Gamma) = Mod(\psi_\emptyset(\psi, \Gamma) \cup \Gamma)$, so that (10.10) follows trivially.

7. $(K \dot{-} \phi) \cap (K \dot{-} \psi) \subseteq K \dot{-} (\phi \wedge \psi)$

   This one requires some further inspection. If we have a consequence α which follows from both $K \dot{-} \phi$ and $K \dot{-} \psi$, then it must follow from $K \dot{-} (\phi \wedge \psi)$:

$$\begin{aligned}(\forall \alpha \in \mathcal{L}_1)[\quad & f^*(g^*(Mod(\psi_\emptyset(\phi, \Gamma) \cup \Gamma))) \subseteq Mod(\alpha) \wedge \\ & f^*(g^*(Mod(\psi_\emptyset(\psi, \Gamma) \cup \Gamma))) \subseteq Mod(\alpha) \implies \\ & \quad f^*(g^*(Mod(\psi_\emptyset(\phi \wedge \psi, \Gamma) \cup \Gamma))) \subseteq Mod(\alpha)]\end{aligned} \tag{10.11}$$

   This is equivalent to the statement that $f^*(g^*(Mod(\psi_\emptyset(\phi \wedge \psi, \Gamma) \cup \Gamma)))$ is contained in the smallest definable set that includes $f^*(g^*(Mod(\psi_\emptyset(\phi, \Gamma) \cup \Gamma))) \cup f^*(g^*(Mod(\psi_\emptyset(\psi, \Gamma) \cup \Gamma)))$. Note that $f^*(g^*(Mod(\psi_\emptyset(\phi \wedge \psi, \Gamma) \cup \Gamma)))$ itself may not be definable.

8. If $\phi \notin K \dot{-} (\phi \wedge \psi)$ then $K \dot{-} (\phi \wedge \psi) \subseteq K \dot{-} \psi$.

$$\begin{aligned} & f^*(g^*(Mod(\psi_\emptyset(\phi \wedge \psi, \Gamma) \cup \Gamma))) \not\subseteq Mod(\phi) \implies \\ & \qquad f^*(g^*(Mod(\psi_\emptyset(\psi, \Gamma) \cup \Gamma))) \subseteq f^*(g^*(Mod(\psi_\emptyset(\phi \wedge \psi, \Gamma) \cup \Gamma)))\end{aligned} \tag{10.12}$$

   If our retraction of $\phi \wedge \psi$ is successful for ϕ, then the results of the retraction will be a subset of the facts obtained by just retracting ψ.

$S_{1,Ab}$ and the AGM Postulates for Revision

We can also examine how our $\psi_+(\phi)$ measures up in terms of the AGM revision postulates, if $K \dot{+} \phi$ is defined to be $\{\alpha \mid \psi_+(\phi, \Gamma) \cup \Gamma \mathrel{|\!\sim} \alpha\}$ or $\{\alpha \mid f^*(g^*(Mod(\psi_+(\phi, \Gamma) \cup \Gamma))) \subseteq Mod(\alpha)\}$, where again $\Gamma = AB(K)$.

1. For any sentence ϕ and belief set K, $K \dot{+} \phi$ is a belief set.

   This trivially holds.

2. $\phi \in K \dot{+} \phi$.

$$f^*(g^*(Mod(\psi_+(\phi, \Gamma) \cup \Gamma))) \subseteq Mod(\phi) \tag{10.13}$$

   Looking back at the definition of $\psi_+(\phi)$ this will hold if $Consis(\phi, \Gamma)$, or if $Mod(\Gamma) \not\subseteq \neg\phi$. Unlike the AGM postulate, ϕ has to be *possible* in Γ before Γ can be revised to entail ϕ.

3. $K \dot{+} \phi \subseteq K + \phi$.

   The conclusions that follow from $\phi \cup \Gamma$ also follow from $\psi_+(\phi, \Gamma) \cup \Gamma$.

$$[f^*(g^*(Mod(\phi \cup \Gamma)))]_\ell \subseteq [f^*(g^*(Mod(\psi_+(\phi, \Gamma) \cup \Gamma)))]_\ell \tag{10.14}$$

   This unravels to:

$$\begin{aligned}(\forall(m_1, m_2))[(m_1, m_2) \in g^*(Mod(\phi \cup \Gamma)) \wedge m_1 \in f([g^*(Mod(\phi \cup \Gamma))]_\ell) \implies \\ (\exists m_2^*)(m_1, m_2^*) \in g^*(Mod(\psi_+(\phi, \Gamma) \cup \Gamma)) \wedge \\ m_1 \in f([g^*(Mod(\psi_+(\phi, \Gamma) \cup \Gamma))]_\ell)]\end{aligned} \tag{10.15}$$

   By the ψ^+ Covering Lemma, $g^*(Mod(\phi \cup \Gamma)) \subseteq g^*(Mod(\psi^+(\phi) \cup \Gamma))$, so (10.15) is true if

$$f([g^*(Mod(\phi \cup \Gamma))]_\ell) \subseteq f([g^*(Mod(\psi_+(\phi, \Gamma) \cup \Gamma))]_\ell), \tag{10.16}$$

   is. (10.16) is easily satisfied if f is monotonic on all inputs.

4. If $\neg\phi \notin K$ then $K + \phi \subseteq K \dot{+} \phi$.

$$f^*(g^*(Mod(\Gamma))) \not\subseteq Mod(\neg\phi) \implies \\ [f^*(g^*(Mod(\psi_+(\phi,\Gamma) \cup \Gamma)))]_\ell \subseteq [f^*(g^*(Mod(\phi \cup \Gamma)))]_\ell \tag{10.17}$$

If it is possible that ϕ can hold in K, then the facts derivable from forcing ϕ to hold are a subset of those from adding ϕ with ψ_+.

5. $K \dot{+} \phi = K_\perp$ iff $\vdash \neg\phi$.

This postulate exemplifies how our framework generalizes AGM. If Γ were entirely soft, that is, every fact were possible, then we would have to interpret the above as

$$f^*(g^*(Mod(\psi_+(\phi,\Gamma) \cup \Gamma))) = Struct(\mathcal{L}_1) \iff \vdash \neg\phi, \tag{10.18}$$

that is, when told to believe a "lie," the shock makes us revert to believing nothing. The flip side of (10.18) is that we can become completely ignorant *only when confronted with such a lie.* If Γ has no hard facts, $g^*(Mod(\psi_+(\phi,\Gamma) \cup \Gamma))$ should return all possible $\mathcal{L}_1$-structures. Then (10.18) will require $f(Struct(\mathcal{L}_1)) = Struct(\mathcal{L}_1)$, which when combined with its properties of contraction and coherence, forces f to be the identity (normal first order logic.)

If Γ contained some hard facts Υ, the result is more confusing. Then intuitively $g^*(Mod(\psi_+(\phi,\Gamma) \cup \Gamma))$ should only return the hard facts of Γ (at least we are shocked, but not enough to despair in all of our facts):

$$f^*(g^*(Mod(\psi_+(\phi,\Gamma) \cup \Gamma))) = Mod(\Upsilon) \iff \vdash \neg\phi. \tag{10.19}$$

This however is a little strange because it requires f to be able to recognize the hard facts of Γ, and pass them through without preferring any of the models. Hence this postulate only seems to make much sense when Γ is entirely defeasible.

6. If $\vdash \phi \iff \psi$ then $K \dotplus \phi = K \dotplus \psi$.

   This again is just showing that equivalent formulas are interchangeable in our paradigm, something that is patently true.

$$(\vdash \phi \iff \psi) \implies f^*(g^*(Mod(\psi_+(\phi,\Gamma) \cup \Gamma))) = f^*(g^*(Mod(\psi_+(\psi,\Gamma) \cup \Gamma))) \tag{10.20}$$

7. $K \dotplus (\phi \wedge \psi) \subseteq (K \dotplus \phi) + \psi$.

$$f^*(g^*(Mod(\psi \cup \psi_+(\phi,\Gamma) \cup \Gamma))) \subseteq f^*(g^*(Mod(\psi_+(\phi \wedge \psi,\Gamma) \cup \Gamma))) \tag{10.21}$$

   This postulate will involve an investigation of how ψ^+ behaves in conjunction with $\wedge$.

8. If $\neg\psi \notin K \dotplus \phi$, then $(K \dotplus \phi) + \psi \subseteq K \dotplus (\phi \wedge \psi)$.

$$\begin{aligned} &f^*(g^*(Mod(\psi_+(\phi,\Gamma) \cup \Gamma))) \not\subseteq Mod(\neg\psi) \implies \\ &\qquad f^*(g^*(Mod(\psi_+(\phi \wedge \psi,\Gamma) \cup \Gamma))) \subseteq f^*(g^*(Mod(\psi \cup \psi_+(\phi,\Gamma) \cup \Gamma))) \end{aligned} \tag{10.22}$$

   If ψ is possible in the result of adding ϕ, then every fact derivable from the addition of ϕ plus the hard addition of ψ is derivable from the addition of $\phi \wedge \psi$.

$S_{1,Ab}$ and the Levi and Harper Identities

The Levi identity (10.1) on its face is not in the spirit of our formalism, because if we just added ϕ to our set of beliefs, it would never be retractable. If on the other hand we added $Ab(\ell) \vee \phi$ after $\psi_\emptyset(\phi \cup \Gamma)$, we would not end up inferring ϕ. The reason why is that our semantics keep track of all of our previous utterances, so although we have retracted ϕ, these previous utterances will react with $Ab(\ell) \vee \phi$ so that it is unlikely that ϕ will hold. (Although we conjecture the cardinality of models which espouse α will increase relative to those which did not!) We are emulating the desired phenomenon discussed in [Friedman and Halpern, 1996], where simply observing ϕ is not enough to conclude ϕ – we have to be told ϕ.

The Harper identity (10.1) is a little more applicable. In this case, if we define $K \dot{+} \phi$ as above, $K \dot{-}_{HI} \phi$ corresponds to adding $\psi_+(\neg\phi, K)$ to K, and then taking the smallest definable set of models which contains both $f^*(g^*(Mod(K)))$ and $f^*(g^*(Mod(\psi_+(\neg\phi, K) \cup K)))$. The only problem is that if $f^*(g^*(Mod(K))) \subseteq Mod(\neg\phi)$, then instead of truly revising K to include models of ϕ along with those of $\neg\phi$, we end up concluding $\neg\phi$, making $\dot{-}_{HI}$ much too weak as a retraction operator.

10.2 Reformulation

10.2.1 Overview of Reformulation

Reformulation focuses on how to change representations so that they are easier to reason about. Related areas of research include *abstraction* and *approximation*. [Giunchiglia and Walsh, 1992] provides a sweeping survey of abstractions, which are defined as mappings from one axiomatic formal system ($\langle \mathcal{L}, \ \mid\!\sim, \ \Gamma \rangle$) to another. They distinguish between theorem-increasing (TI), theorem-decreasing (TD) and theorem constant (TC) abstractions, which describe how the mapping preserves theorems in one system in the transition to the other system. The work shows how a large array of problems in AI fall into their framework, which we do not repeat here.

But the attraction of reformulation is not restricted to AI. Users of databases also find it helpful to reformulate their databases, to optimize them it the presence of some set of expected queries. [Chirkova, 2000] finds rewritings of databases which are at most a linear multiple of the original, which can answer a set of queries in a shorter amount of time.

10.2.2 Elaboration Tolerance Versus Reformulation

The field of reformulation, like belief revision, looks very much like elaboration tolerance, as it also involves change to a knowledge base of some sort. However, the underlying goals are different and intensions are very different. Reformulation seeks to find ways to change representations so that it is easier to solve some particular problem. Usually the change loses information, or makes some particular structure more perspicuous, to reduce computation. On the other hand, the purpose of an elaboration is not to change the (syntax of the) representation for computational means, but to change its meaning.

Furthermore, while reformulation is concerned with rewriting a *particular* representation to make it more efficient, elaboration tolerance cares more about the efficiency with which a representation can admit change, *generally*. Elaboration tolerance looks for *one* particular representation that is amenable to *any* proposed "reformulation." The ultimate elaboration tolerant representations will not require any change to their language; at most they will simply require the addition of more symbols to express new concepts (as in human discourse).

10.3 Bayesian Networks

When we speak of a Bayesian network's elaboration tolerance, we refer to the ability to add more dependencies between variables, simply by adding nodes, with minimal effort. The minimal effort stipulation requires us to only add child nodes to an existing network; adding parents to a node will require us to change the associated conditional probability tables. Adding a child however is no problem, as this does not require any changes to the original structure.

While primary dependencies can only be extended from a node to its child, *secondary dependencies* can occur between two nodes, even those that are not directly connected to each other. This phenomenon is based on the concept of the *d-separation* of nodes: knowing the value of the ancestor or descendant of a node, can affect the values the node can take. More importantly, two nodes can *become* correlated if a mutual child's value is known. If the original Bayesian network accurately represents the fundamental interactions between components, these secondary dependencies that follow from the model should be representable as well.

A Bayesian network also exhibits elaboration tolerance through the use of its *exogenous* variables. These are input variables that influence the behavior of other nodes in the network, but are not considered to be explained by the network. These exogenous variables could serve as easy entry points ("back doors") to add more parents/influencers to a node when desired, to better elaborate some aspect of probabilistic reasoning. These exogenous variables play a similar role to the *Ab*s we have used in our logical representations.

Finally, [Pearl, 1988] notes that probabilistic systems are inherently intensional. Intensional systems are more "Gestalt," in that a particular unit of meaning depends intrinsically on other parts of the system. We conjecture that intensional systems will be more elaboration tolerant than extensional ones, as they intrinsically contain more of the inferential mechanisms to properly constrain the relationships between various units of meaning. Extensional representations on the other hand must explicitly model all of these relationships. This leads to more data that must be examined and altered in the face of an elaboration. [1]

10.4 Neural Networks

In Chapter 3 we argued that an obvious connection between the symbols of the representation, and the semantics is important for a representation's elaboration tolerance. [McCarthy, 2002b] notes that neural networks are not particularly elaboration tolerant. Consider the neural network NETtalk [Sejnowski and Rosenberg, 1988] used for text-to-speech synthesis, and consider a change where we pronounce the letters "x" (*"sh"*) and "q" (*"ch"*) as Chinese rather than English symbols. A person will only need to be told these two alterations, while NETtalk would have to revise on the

[1] For a brief examination of the difference between intensional and extensional systems, see the beginning of [Perez and Jiroušek, 1985].

order of 18,629 weights, in a manner that requires massive computation (back propagation). NETtalk is not elaboration tolerant as per our discussion in Section 3.2, as not only are the elaborations not of less complexity than the representation, but they are not easily predictable – it is not clear what weights need to change in what way to achieve the desired behavior.

However this example does reveal another issue, which is the modality with which we can change a representation. We could argue that an extended implementation of NETtalk, with two extra inputs that can switch the pronunciations of "x" and "q," is elaboration tolerant. However, elaboration tolerance is a property that holds over *all possible changes*, and not a particular one. A version of NETtalk which had one input used to toggle every possible elaboration in this domain would be intolerably large and complex, except perhaps for the simplest (finite?) domains.

Compare this proposed formalism to our $S_{1,Ab}$, which can represent any change just by addition of the proper formula! The expressivity of the representation, in the sense of whether it can represent meta-operations, does effect the ease of elaboration tolerance. Neural networks do not have this modality (perhaps unless they were allowed to have feedback loops).

Chapter 11

Elaboration Tolerance and AI

In this chapter we discuss some of the ramifications of elaboration tolerance. The relevance of elaboration tolerance to the future of the logical AI program is discussed in Section 11.1. We make some correspondences between mechanizing elaborations and the frame problem in Section 11.2. Section 11.3 attempts to disambiguate between intensional and extensional representations, mentioning how intensional ones appear to be primed for elaboration tolerance. We conclude (Section 11.4) with a discussion of the relationship between elaboration tolerance and nonmonotonic reasoning, and how it is possible that the former is actually the cause for the latter, instead of the other way around.

11.1 The Future of Logical AI

[McDermott, 1987] asserts that the logicist tradition of explicitly writing down all knowledge is doomed. One of McDermott's reasons is that that there is no clear sense of when we have written down all common sense knowledge. McDermott notes that "all the straightforward inferences follow from the axioms that have been written down." Cyc [Cyc, 2003] has been criticized by many for failing to capture the whole of common sense reasoning in logic. Expert systems, once the bread and butter of AI, are now largely abandoned.

Part of the reason why logical AI has not fared as well as it should is because of a view that knowledge is static and completely describable. The mistake is in assuming that some static knowledge representation will be able to represent the whole of common sense knowledge. Common sense reasoning is inherently inexact and incomplete; new "axioms" are discovered by enterprising humans every day! Furthermore, it is not possible to ever complete the whole of human knowledge; there are always circumstances which require more details, and retractions of the facts we believe, as explained previously at the end of Section 1.1. It is a fallacy to assume we can codify all human reasoning, once and for all.

A workable AI framework must have *built within it* the ability to accept new information and retract inconsistencies as necessary. It has to be always extensible. Instead of formulating "pompous" knowledge representations which believe they know everything and are always correct, we want "meeker" representations, always doubting, always ready to accept elaborations of present facts. [1] More formally, our framework should be able to represent a bit of knowledge, but then immediately and easily extend to a more detailed representation which can embody new facts while consistently retaining compatible facts from its previous incarnation. This is the whole point behind the missionaries and cannibals [McCarthy, 1997] and egg cracking [Morgenstern, 1998] problems, where our representation is encouraged to be able to tolerate elaborations such as there being a bridge, or cracking an ostrich egg.

11.2 The Frame Problem

Most abstractly, we can characterize our study of how elaborations affect our representation in terms of action theories: the elaborations correspond to actions, and the representation is the state of the world we want to describe. This intuition is reflected in our desire that formulas should not be affected by an irrelevant elaboration, i.e., a representational version of the *frame problem.*

The intuitions become even more interesting when we consider the effects of our *epistemic entrenchments* – when a change is required to be made to our knowledge base, the epistemic entrenchment helps decide which, amongst possible changes, are

[1] They always said that the meek would inherit the earth.

the most likely to be made. This parallels static constraints, which constrain the trajectory of our system as time progresses. Furthermore, they can also give direction as to which change is preferred – [McIlraith, 2000] show how a causal interpretation of material implication used in ramifications is all that is needed to decide how changes play out.

As we have mentioned many times before, we desire our representation to be "Cartesian," where interactions between facts are minimal, so that it is easier for us to forecast our elaborations and guarantee that the results are as expected. This notion of a basis set of fluents has popped up in many solutions to the frame problem, as shown in [Shanahan, 1997].

We have the same problem with elaborations – we want to make sure that our changes affect the fluents in the correct way. For example, changing the preconditions on rowing should not affect whether there is a bridge, but it should affect the solution of the problem. Hence both elaboration tolerance and theories of action are in some sense heavily reliant on a representation that partitions facts so that they interact only in understood pre-determined ways. The altered epistemic entrenchments of the form $Ab(\ell_k) \implies Ab(\ell_j)$ described at the end of Section 9.6 comprise a first stab at this idea.

11.3 Intensional versus Extensional Aspects

Natural language is inherently *intensional.* Its property of being able to assert facts without having to tell the whole story seems to be precisely what makes it so elaboration tolerant. *Extensional* systems on the other hand do not model meanings with symbols as well, and thus require more infrastructure in order to tolerate elaborations. Part of the reason for this disparity is that while extensional systems treat the meanings of symbols as fixed, intensional ones can change as new information is added (consider probabilistic systems).

For extensional systems to compete, they will need a kind of "entry point" by which to inject elaborations. This is evident in the use of exogenous variables within Bayesian Networks as well as the introduction of *Ab* to first-order logic sentences augmented with circumscription as an inference mechanism. In the second case, we are using *Ab*s to partially reify our formulas, to bring us closer to an intensional kind of representation.

We can examine the difference on another scale. Consider the rule

$$has(coffee, sugar) \implies tastes\_fine(coffee). \tag{11.1}$$

Statements like (11.1) comprise extensional systems because they specify that as long as $has(coffee, sugar)$, then $tastes\_fine(coffee)$, without regard to other factors such as $has(coffee, diesel\_oil)$ which may negate $tastes\_fine(coffee)$ [van Benthem, 1988]. In order for these factors to be represented, the rule above must be refined and those factors explicitly stated. Intensional systems such as probabilistic statements intrinsically are capable of stating not only what usually holds, but can allow the inference to change in the presence of updated information. Hence, we can assert without contradiction $P(tastes\_fine(coffee) \mid has(coffee, sugar)) = .9$ and $P(tastes\_fine(coffee) \mid has(coffee, sugar), has(coffee, diesel\_oil)) = 0$, whereas with the rules we cannot properly assert the dependence of $tastes\_fine(coffee)$ on $has(coffee, sugar)$ without talking about $has(coffee, diesel\_oil)$, unless we use non-monotonicity.

Intuitively speaking, intensional systems allow for a "background aether" to be embedded within the representation to allow us to change our statements as necessary. With extensional systems, on the other hand, there is no aether, only a vacuum, and what you see is what you get – the symbols encode all the meaning and leave no space for exceptions, new facts, etc.

[Pearl, 1988] has this same intuition, asserting that "extensional systems" are modular, in that a rule of the form $A \rightarrow B$ means assert B when we see A, regardless of the other facts known. But the mass of human knowledge is inherently un-modular, in that a fact will depend on a multitude of facts, which are not even all known. When human knowledge is encoded in such forms, these myriad relationships are lost. The reason is because containing all the exceptions, even if possible, would clutter our statements and make them computationally (and representationally) useless. Instead we attach a special semantics to such rules, inherently adding a "unless otherwise," when A is observed assume B. Human language implicitly assumes that almost all utterances are open to elaboration/are inherently incomplete, and because of this built-in assumption, can tolerate elaborations. In order to make our representations have these same properties, we will have to build in this notion of defeasibility, as accomplished in our system $S_{1,Ab}$.

11.4 Nonmonotonic Reasoning

[McCarthy, 1997] notes that "Elaboration tolerance clearly requires nonmonotonic reasoning", but one may argue that elaboration tolerance is what has generated the need for nonmonotonic reasoning. The original impetus for nonmonotonic reasoning, "Tweety is a bird; therefore Tweety flies" had to be nonmonotonic just so that it could *tolerate* the elaboration that Tweety is a penguin. It is evident in many approaches to the frame problem [Shanahan, 1997], and other AI formalisms, as well as human discourse, that addition of facts is the primary means of changing representations, which will require nonmonotonic consequence relations.

In fact, viewing some of these representational problems from a perspective of maximizing elaboration tolerance, instead of properly utilizing nonmonotonic reasoning, can provide some fresh, intriguing insights. For example, [McCarthy, 1986] shows how to write theories so that they are amenable to nonmonotonic reasoning. One of the examples given is formalizing whether birds can fly.

First, we include the entire theory below, slightly altered to fit our paradigm:

$$
\begin{aligned}
&(\forall x)[\neg Ab(1,x) \implies \neg Flies(x)]\\
&(\forall x)[Bird(x) \implies Ab(1,x)]\\
&(\forall x)[Bird(x) \wedge \neg Ab(2,x) \implies Flies(x)]\\
&(\forall x)[Ostrich(x) \implies Ab(2,x)]\\
&(\forall x)[Penguin(x) \implies Ab(2,x)]\\
&(\forall x)[Ostrich(x) \wedge \neg Ab(3,x) \implies \neg Flies(x)]\\
&(\forall x)[Penguin(x) \wedge \neg Ab(4,x) \implies \neg Flies(x)]\\
&(\forall x)[Bird(x) \wedge \neg Ab(5,x) \implies Feathered(x)]\\
\\
&(\forall x)[Ostrich(x) \implies Bird(x)]\\
&(\forall x)[Penguin(x) \implies Bird(x)]\\
&(\forall x)[Canary(x) \implies Bird(x)]
\end{aligned}
\tag{11.2}
$$

McCarthy's formalism asserts $Ab(aspect_n(x))$ instead of our $Ab(n,x)$ to assert that object x is abnormal in aspect n. $aspect_n(x)$ is used to distinguish the fact that objects can be abnormal in different ways, and will affect other fluents differently. For example, aspect 1 above in the abnormality $Ab(1,x)$ is used to determine whether an object can fly. $Ab(2,x)$ on the other hand, is used to label that it is strange when an object (namely, a bird like an ostrich or a penguin) *does not* fly.

[McCarthy, 1986] circumscribes the theory in (11.2), minimizing Ab and varying only the predicate $Flies$. $Flies$ is allowed to vary because "the purpose of the axiom set is to describe what flies." This allows us to conclude that the only objects which fly are birds which are not ostriches or penguins.

McCarthy further notes that allowing $Bird$, $Ostrich$, $Penguin$, and $Canary$ to vary as well will lead to an empty extension of Ab, as well as empty extensions for $Bird$, $Flies$, $Ostrich$, $Penguin$, and $Canary$. In short, under this circumscription policy, there are no birds, of any kind, and nothing flies. If witnesses such as

$$Canary(Tweety) \wedge Ostrich(Joe), \tag{11.3}$$

are added, then circumscription where Ab is minimized and all the predicates varied will lead to the expected conclusion that $Tweety$ flies but Joe does not.

It is distressing that there is no obvious means to discover a circumscription policy that will entail the correct conclusions. [McCarthy, 1986] gives a partial solution in terms of *policies*, where the set of predicates to be minimized and varied is explicitly listed, but in no way helps us decide *how to construct such policies.*

It is intriguing to instead look at this same problem through the lenses of elaboration tolerance. [McCarthy, 1986] views the axiomatization in terms of constructing the right axioms based on Ab and some policy so that the minimizations lead to the correct nonmonotonic conclusion. Consider if we instead viewed the axioms in (11.2) as the remnants of conversational rules, where the amount that needed to be said is minimized (as speakers are lazy), while the number of pertinent facts uttered is maximal (as speakers, in order to maintain their laziness, must be efficient with their utterances). We can view the mechanism of nonmonotonicity solely as a means of filling in the gaps between the speaker's utterances, a set of communication convention defaults that allows both speaker and listener to greatly abbreviate their utterances without loss. The Abs in (11.2) are used to indicate where facts are incomplete, and later prone to elaboration. (We can think of the Abs as "entry points" for inserting more preconditions on a rule later on.)

We can reinterpret (11.2) in terms of the simple framework used in Section 4.4 to motivate our pursuit of additive elaboration tolerance. First off, we see that the last three axioms in the theory correspond to hard facts, as they omit Ab. The first eight axioms are soft truths, rewritten as:

$$\begin{aligned}
(\forall x)[Ab(1,x)\vee\ & (\neg Flies(x) \wedge \neg Bird(x))] \\
(\forall x)[Ab(2,x)\vee\ & ((Bird(x) \implies Flies(x))\wedge \\
& \neg Ostrich(x)\wedge \\
& \neg Penguin(x))] \\
(\forall x)[Ab(3,x)\vee\ & (Ostrich(x) \implies \neg Flies(x))] \\
(\forall x)[Ab(4,x)\vee\ & (Penguin(x) \implies \neg Flies(x))] \\
(\forall x)[Ab(5,x)\vee\ & (Bird(x) \implies Feathered(x))]
\end{aligned} \tag{11.4}$$

Now these axioms are different from the ones we introduced in Section 4.4, as our *Ab*s are parameterized not only over the labels 1, 2, 3, ..., but over objects in the domain of the birds theory as well. We can compare this to our construction in Chapter 6, where *Ab*s may only take labels as arguments. The notation in (11.4) is more general, and suggests that our separation of formulas might be too severe. Allowing our *Ab* labels to be parameterized in this way is a topic we note for future research.

Moving on, [McIlraith, 2003] notes the constants 1, 2, 3, ... serve as *contexts*, grouping sets of related axioms. Aspect 1 regulates both whether something is a bird and whether it flies. In terms of our framework, by default we are to assume that each individual object is not a bird and does not fly. The second axiom in (11.4) asserts three facts: by default each individual bird flies, and by default objects are neither ostriches or penguins. As we observed in Example 9.4.1, these three facts are correlated – if something happens to make one of them false, the other two will be "retracted" automatically. The reason is that $Ab(2, x)$ will have to hold, and thus their truth will not matter anymore. It is interesting to see how the correlation derived with respect to elaboration tolerance relates to McCarthy's original intentions when writing these axioms. For example, if a bird does not fly, we become agnostic about whether it is not an ostrich, or penguin. Conversely, if a specimen is an ostrich, we become agnostic about whether it flies or not. The last three axioms express three unrelated soft truths: ostriches do not fly, penguins do not fly, and birds have feathers.

As a brief aside, note that the axioms in (11.4) is different from the alternative version:

$$\begin{aligned}
(\forall x)[\mathbf{Ab(1)}\vee\ & (\neg Flies(x) \wedge \neg Bird(x))] \\
(\forall x)[\mathbf{Ab(2)}\vee\ & ((Bird(x) \implies Flies(x))\wedge \\
& \neg Ostrich(x)\wedge \\
& \neg Penguin(x))] \\
(\forall x)[\mathbf{Ab(3)}\vee\ & (Ostrich(x) \implies \neg Flies(x))] \\
(\forall x)[\mathbf{Ab(4)}\vee\ & (Penguin(x) \implies \neg Flies(x))] \\
(\forall x)[\mathbf{Ab(5)}\vee\ & (Bird(x) \implies Feathered(x))]
\end{aligned} \tag{11.5}$$

where the parameterization within the *Ab*s has been removed. (11.5) asserts that by default each of these facts are true, and we can retract *all instances* by asserting one atom $Ab(n)$. The approach in [McCarthy, 1986], that we have depicted in (11.4) gives finer-grained control over the axioms, allowing the retraction of an *instance* of the axiom, rather than the entire rule.

Applying our version of circumscription to (11.4) along with the hard truths of (11.2), where all symbols are varied and Ab is minimized, would result in there being no birds, ostriches, and penguins, and nothing would fly, as noted already in [McCarthy, 1986]. But from the point of the view of our soft truths, which should be interpreted as true whenever possible, this is precisely what should be inferred. When the witnesses $Tweety$ and Joe are added, the minimization should deduce the proper conclusions for each, but still assume that nothing else flies, is a bird, etc. [2]

It is important that the last three axioms of (11.2) are hard truths, ensuring that $Tweety$ and Joe fall into their proper categories. Since these hard truths lack Abs, they are immune to the pressure from our circumscriptive force. The example in Section 11 of [McCarthy, 1986] shows what happens if this were not the case. Consider the theory

$$\begin{aligned}
&(\forall x)[Ab(1,x) \vee (\neg Flies(x)) \wedge \neg Bird(x)] \\
&(\forall x)[Ab(2,x) \vee (Bird(x) \implies Flies(x))] \\
&(\forall x)[Ab(3,x) \vee (Canary(x) \implies Bird(x))]] \\
&Canary(Tweety)
\end{aligned} \tag{11.6}$$

Here, the rule $(\forall x)[Canary(x) \implies Bird(x)]$ has been softened. This allows a break in the reasoning, where we cannot infer whether $Tweety$ flies or not, because we cannot be sure that the inheritance rule $Canary(Tweety) \implies Bird(Tweety)$ applies, as it conflicts with our first default assumption that most things are not birds and do not fly. McCarthy's solution here is to introduce *prioritized circumscription.* We believe a simpler solution lies in formulating *epistemic entrenchments* within the

[2]Note that the axiomatization in (11.5) would result in similar conclusions, under the aforementioned circumscription. Yet when $Tweety$ and Joe are added, we will also get the correct conclusions about whether they fly, etc., but will instead remain agnostic about whether anything else flies, is a bird, etc. These atypicalities serve to instantly render our rules inapplicable. Thus we see the wisdom of parameterizing the soft truths with respect to individuals.

language, say by adding:

$$(\forall x)[Ab(3,x) \implies Ab(1,x)] \tag{11.7}$$

If we let all predicates vary, then the expected conclusion is that $Bird(Tweety) \wedge Flies(Tweety)$, as (11.7) expresses our desire that the truth of the third axiom overrides that of the first. Note that these epistemic entrenchments are implemented within the framework of normal circumscription, without requiring any additional expressive power.

11.5 Computability

We briefly indicated in Section 3.4.3 that the *synthetic* approach to writing predicates, while elaboration *in*tolerant, appears to be more computationally practical than the *analytic* forms. Since the analytic form *ascribes* properties and relationships to some key-like object r, more things can be said (or left unsaid) about r. Synthetic syntax on the other hand of the form $r = f(\overline{x})$ asserts that r is the sum total of all features of x, and nothing else.

As a more concrete example, consider the simple task of inference. Say we want to know if a row action occurred at time t_0 with $cannibal1$ participating. If we used the synthetic syntax to encode our information, we would simply have to search for instances of $Row(cannibal1, *, *, *, t_0)$ in a giant table. This operation is much faster and less complex than searching for an object r with the properties of $Rowing(r) \wedge Rower(r, cannibal1) \wedge Time(r, t_0)$. Intuitively, the analytic form is elaboration tolerant precisely because *it allows facts to be left unsaid.* The synthetic form on the other hand, must bundle up all related information all at once. [3]

But because the analytic form remains agnostic on certain facts, we need more powerful reasoning in order to fill in these unknown facts as necessary. For example, consider the following representational problem pointed out by [Fikes, 2003a]. We

[3] We could assert synthetic facts of the form $(\exists m)Row(cannibal1, m, bank1, bank2, t_0)$ to defer having to assert all facts at once, but this form still forces us to assert that there is some object m participating in the Row, which may already be asserting too much.

want to express the fact that C_1 is equal to $C_2 \cup\ C_3 \cup C_4$ using RDF, a schema that requires all facts to be expressed as triples.

We must assert something like:

$$(\textit{union-of}\ C_1\ (\textit{list}\ C_2\ C_3\ C_4)), \tag{11.8}$$

where *list* is defined in terms of standard car/cdr notation as:

$$(\textit{list}\ C_2\ C_3\ C_4) =_{def} (\textit{cons}\ C_2\ (\textit{cons}\ C_3\ (\textit{cons}\ C_4\ \textit{nil}))), \tag{11.9}$$

resulting in the statement

$$(\textit{union-of}\ C_1\ (\textit{cons}\ C_2\ (\textit{cons}\ C_3\ (\textit{cons}\ C_4\ \textit{nil})))). \tag{11.10}$$

This method of representation clearly has computational issues. (It takes $O(n)$ to find out if one set is a part of the union of C_1, where intuitively it should take $O(1)$, since C_1 should "know" what sets compose it). More importantly, it is not the most elaboration tolerant form possible. The analytic form of this sentence would be something like:

$$\begin{array}{c}(\textit{composes}\ C_2\ C_1)\\(\textit{composes}\ C_3\ C_1)\\(\textit{composes}\ C_4\ C_1)\end{array} \tag{11.11}$$

This would allow us to easily change what sets are part of the union – simply add or delete sentences of the above form. However, a stronger closed world assumption is then required to conclude certain important facts, such as the fact that C_2, C_3 and C_4 are the *only* subsets of C_1. The more synthetic form in (11.10) does not need this extra machinery. Hence we suspect that elaboration tolerant structures will require more powerful consequence relations in order to "fill in the blanks" properly.

Chapter 12

Future Research Directions

There are quite a few avenues for future research from here. As we mentioned before, $S_{1,Ab}$ was only constructed to pose as a witness for additive elaboration tolerance. We believe more intricate structures can be constructed using the principles we have demonstrated here. Section 12.1 evaluates this idea with respect to adding more expressive elaborations.

Another future research direction which is crucial is a *working implementation* of these ideas. Section 12.2 sketches out a possible system.

There are other avenues for research that hail from other fields that could prove enlightening. Just as database design schemas have provided the foundations for constructing design principles for elaboration tolerance, we believe the notion of database indexing (Section 12.3) could play a role in evaluating our representations. An elaboration tolerance *Advice Giver* is only a nice fruit of our labors from Chapter 3, described in Section 12.4. We also believe it is important to study probabilistic systems (Section 12.5), as they seem to embody many of the features we desire for elaboration tolerance, without running into issues of undecidability. Finally, as we mentioned at the beginning of this thesis, elaboration tolerant structures will provide a platform from which we can study the more advanced concepts of *approximation* [McCarthy, 2000].

12.1 Representing Other Elaborations

We have focused on how to add formulas to retract and add formulas to our axiomatization, but we have not addressed other elaborations, such as actually adding new soft and hard facts to our axiomatization. We have shown how to implement some of these elaborations using the formulas of $Elab^+(\mathcal{L}_1, \mathcal{L}_{Ab})$:

1. To add a soft fact ϕ we add $Ab(\ell) \vee \phi$ to our axioms.

2. To add a hard fact ψ, we simply add ψ to our axioms.

3. We can change parameters by adding the formula $\psi_+(x = n, \Gamma)$, where x is some constant and n some new value.

There are other elaborations that could implement some of the possibilities in [McCarthy, 1997], but for which we will need a more expressive framework:

1. We can add a precondition to a soft fact $Ab(\ell) \vee \phi$ by adding $\neg Ab(\ell') \wedge Prec \implies \neg Ab(\ell)$. This can be rewritten as $Ab(\ell) \wedge Prec \implies Ab(\ell')$, to show that it is a generalization of ψ_+, with $Prec = \ell' < \ell$.

   It could also be rewritten to the form $Prec \implies (Ab(\ell) \implies Ab(\ell'))$. Under this interpretation, we see that when $Prec$ holds, label $Ab(\ell)$ outranks $Ab(\ell')$.

2. As shown in Section 11.4, we can extend our Abs to also map over elements of S_1. Elaborations could be applied on *instantiations* of sentences of the form $Ab(\ell, x) \vee \phi(x)$.

3. We could construct a more flexible epistemic entrenchment as mentioned in Section 9.6. The idea is to remove the ordering relation $<$ in S_{Ab} and explicitly determine an ordering by asserting $Ab(\ell_k) \implies Ab(\ell_j)$ instead of our "temporal" $\ell_k > \ell_j$. If we could add statements of these forms as elaborations, not only could we construct our own prioritizations over formulas, but this prioritization could itself be amended.

   This flexibility however may outweigh the benefits, because now we will have to specify every link of the precedence. There is also some intolerance in the sense that once a precedence is set, it cannot be retracted. One possible way out of this problem is to utilize the system $S_{1,Ab,Ab} = S_{Ab} \cdot S_{Ab} \cdot S_1$ to create retractions over retractions.

12.2 Implementation of Elaboration Tolerant Systems

This thesis has two main contributions: a framework and design principles for constructing elaboration tolerant structures in general, and a study of the concept of additive elaboration tolerance. One criticism of this work is that we have designed "castles in the air," by formalizing the concept of elaboration tolerance in essentially second-order logic, which is highly undecidable. In particular, [McDermott, 1987] points out that nonmonotonic formalisms suffer from the "You can't find out" and "You don't want to know" issues, where it may be undecidable to find nonmonotonic consequences, and the conclusions resulting from nonmonotonic reasoning are often too weak to be helpful. While we have proven that our retraction and addition operators have the desired effect, we have not given any indication of what other formulas stay true after the elaboration, or if we can even prove what remains the same.

One way to counter these criticisms is to develop a working system with additive elaboration tolerance. One promising candidate is *answer set programming*, a clean formalization of logic programs, that also has some very fast implementations (see `lparse` [Syrjänen, 2000] and `smodels` [Simons, 2000]). Since it is decidable, of course we will not have the expressivity we desire. In order to represent $\psi_\emptyset$, we not only need to be able to express ϕ, but also $\neg\phi$. Since true negation over arbitrary formulas is not easily accomplished in logic programming, we will have to restrict the form of ϕ that is expressible. One possibility is to let ϕ be of the form $P_1(x) \wedge \ldots \wedge P_n(x) \implies Q(x)$, where the P_i and Q are atoms. Given this restriction, our building blocks for possible formulas are of the form:

$$
\begin{aligned}
&Ab(\ell) \vee (P_1(x) \wedge \ldots \wedge P_n(x) \implies Q(x)) \\
&Ab(\ell) \vee (P_1(x) \wedge \ldots \wedge P_n(x) \wedge \neg Q(x)), \\
&\textit{and} \\
&Ab(\ell) \wedge x < \ell \implies Ab(x)
\end{aligned}
\tag{12.1}
$$

representable as logic program statements of the form:

$$
\begin{aligned}
&\mathtt{Q(x)\ :\text{-}\ not\ Ab}(\ell), \mathtt{P_1(x)}, \ldots, \mathtt{P_n(x)} \\
&\mathtt{P_1(x)}, \ldots, \mathtt{P_n(x)}, \neg\mathtt{Q(x)\ :\text{-}\ not\ Ab}(\ell) \\
&\textit{and} \\
&\mathtt{Ab(x)\ :\text{-}\ Ab}(\ell), \mathtt{x} < \ell
\end{aligned}
\tag{12.2}
$$

Showing that these kinds of formulas will have the desired results is a topic for future research. Note that we could in this case go ahead and generalize the *Ab* labels to also parameterize over constants to get added expressivity.

12.3 Database Indexing

The methods of database indexing may be relevant to our study of how to make elaboration tolerant representations [da Silva, 2003]. Database indexing reflects the organization of information – a simple, unencumbered index implies that the data is is arranged similarly. If an index becomes too large, this indicates that the data needs to be organized further. If two indices refer to the same data, then this could imply multiple meanings for a particular formula, which in general will muck up elaboration tolerance. If a formula is not referred to by any index, this means it is either irrelevant or some meaning is not being properly represented.

The computational complexity of the time it takes to use an index could provide a metric for the complexity of the accompanying representation.

12.4 An Elaboration Tolerance Advice Giver

Chimaera [McGuinness et al., 2000] is a system used to create and merge ontologies. It also can diagnose them, by detecting abnormalities in axioms. [Fikes, 2003b] points out that a similar device could be built with respect to elaboration tolerance. We could use the elaboration tolerance-promoting principles discovered in Chapter 3 as a framework to analyze arbitrary representations. The system could be so powerful as to devise the intended semantics of the representation from the syntax, and based on this, give advice on how to rewrite the representation to make it more elaboration tolerant.

12.5 Probabilistic Systems

Probabilistic systems are a monotonic formalism that can be easily elaborated. Consider the statements

$$\begin{aligned} Pr(\phi \mid \psi) &= 1 \\ Pr(\phi \mid \psi, \alpha) &= 0, \end{aligned} \tag{12.3}$$

While at first we believe something like "$\psi \implies \phi$", after hearing the elaboration α we instead infer "$\psi \implies \neg\phi$." Note that these two statements are entirely consistent. Furthermore, these statements are expressed in a monotonic logic (they only require interpretation of the arithmetic operator over the reals and Bayes rule). [1]

In fact we can perpetually elaborate the value of concept X_1 as shown below:

[1] These two statements are comparable to Reiter's default logic axioms [Reiter, 1980]:

$$\frac{\psi : \neg\phi}{\phi}, \frac{\psi, \alpha :}{\neg\phi} \tag{12.4}$$

$$\begin{aligned}\{\quad & Pr(X_1) = p_1, \\ & Pr(X_1 \mid X_2) = p_2, \\ & Pr(X_1 \mid X_2, X_3) = p_3, \\ & \ldots, \\ & Pr(X_1 \mid X_2, \ldots, X_n) = p_n\},\end{aligned} \tag{12.5}$$

provided each p_i remains in the *open* interval $(0, 1)$. (If we ever set one of the p_i to 1 or 0, we are essentially turning it into a hard truth/falsehood, from which there is no recovery. Forcing $0 < p_i < 1$ keeps the relationships soft and therefore mutable.)

How is this elaboration tolerance, without nonmonotonicity, possible? The obvious answer is that the nonmonotonicity is expressed one level deep, *within* the logic. α, ϕ, and ψ are never outright asserted as being true or false by the statements of $\mathcal{L}_{prob}$. Quite the opposite, they are simply *terms* about which relationships between them are uttered, very much in the analytic style. Our statements stop short of ever ascribing any particular truth value to the propositions.

The way in which probabilistic statements embed elaboration tolerance inside a monotonic formalism is intriguing and deserves further examination. We believe this is due to the underlying semantics: probabilities derive their meaning from sample spaces; a formula ϕ's probability given ψ is defined to be the percentage of the sample space of where ψ holds where ϕ is also true. Adding statements of the form $Pr(\phi \mid \psi, \tau) = p'$ is almost always consistent (exceptions are when some probability is 0 or 1), because the statements of probability constrain the remaining sample space in such a way that there is always enough "space" for subsequent statements.

By "space" we mean the following concept. Each probability statement of the form $Pr(\phi \mid \psi) = p$ does not rule out, or cut out, parts of the sample space (unless $p = 0$ or $p = 1$). Instead, it constrains the relative sizes of subspaces: in this case the space where $\phi \wedge \psi$ is true must occupy p percent of the space where ψ holds. The elaborative statement $Pr(\phi \mid \psi, \tau) = p'$ will then assert that the ratio of the space where ϕ, ψ, and τ all hold to the space where $\psi \wedge \tau$ hold must be p' – this does not affect our constraint that $Pr(\phi \mid \psi) = p$ in the slightest. When $p = 0$ or $p = 1$, a part of the sample space does disappear, and no subsequent probabilistic statement can ever "bring it back."

It is illuminating to compare this probabilistic semantics with our discussion of semantics in Chapter 3. In both cases we envision a universe of all possible states of the world, and our semantics relies on the likelihood/existence of all of these possible states. Any logic-based semantics with a monotonic consequence relation will remove models as more statements are added. Once we assert α, all those models which disagree with α are permanently removed.

Compare this subtraction of models for logical semantics with our probabilistic semantics. Unless a probability is 0 or 1, no part of the sample space is ever removed. Instead the relative percentages of sample spaces are simply fixed. If our elaborations follow the form of the statements in (12.5), then all we are doing is constraining the percentage of X_1 overall, then the percentage of X_1 in X_2, and then the percentage of $X_1 \wedge X_2$ lying in X_3, and so forth. All we do is constrain the ratio of spaces between subsets, possible because our space is "fine" enough to be split up to maintain arbitrary proportions.

The one catch with the probabilistic approach is that it is descriptive, but not declarative. We know how our degree of belief changes as we learn various facts, but we do not know which facts we believe *currently*, or are most likely. This will require some computation. Or conversely, given a set of facts that we do believe to hold, we will have to do mathematical computation in order to work out the likelihood of such facts. This computation is probably analogous to the nonmonotonic reasoning required for us to come to conclusions in our logical framework. The difference is that the first is decidable, while the second, not necessarily so. Probably the reason for this disparity is that the probabilistic semantics works only over the restricted set of propositional atoms.

12.6 Approximate Objects and Elaboration Tolerance

As mentioned in Section 1.1, using approximate objects in our representations will require some kind of elaboration tolerance by their nature. In fact, elaboration tolerant representations may be absolutely necessary to properly represent approximate objects and theories, since by nature they are those concepts which are intrinsically elaboratable.

Appendix A

Proofs

A.1 Proof of Proposition 5.3.1

Proposition 5.3.1 [Properties of $\mid\!\sim$ defined by f]. *Say* $\Gamma \mid\!\sim \phi \iff f(Mod(\Gamma)) \subseteq Mod(\phi)$. *Then:*

1. *$\mid\!\sim$ obeys right monotonicity and right conjunction.*

2. *If f obeys contraction then $\mid\!\sim$ obeys inclusion.*

3. *Say $f(X)$ is defined to be of the form*

$$f(X) =_{def} \{x \in X \mid (\forall y \in X)[R(x,y)]\}, \tag{A.1}$$

   for some relation R. Then f satisfies contraction, coherence, and left disjunction.

4. *Say $f(X)$ is defined to be of the form*

$$f(X) =_{def} \{x \in X \mid (\forall y \in X)[\neg y < x]\} \tag{A.2}$$

   where $<$ is well-founded and transitive (no infinitely descending chains). Then $\mid\!\sim$ satisfies cautious monotonicity.

Proof. 1. Right monotonicity: Assume $\Gamma \mid\!\sim \phi \wedge \phi \vdash \psi$. In terms of f this means:

$$\begin{aligned} & f(Mod(\Gamma)) \subseteq Mod(\phi)) \\ & Mod(\phi) \subseteq Mod(\psi) \end{aligned} \tag{A.3}$$

and we want to show that $f(Mod(\Gamma)) \subseteq Mod(\psi)$. But this is straightforward, given (A.3).

Right conjunction: this time we assume

$$\begin{aligned} & f(Mod(\Gamma)) \subseteq Mod(\phi) \\ & f(Mod(\Gamma)) \subseteq Mod(\psi) \end{aligned} \tag{A.4}$$

And want to show that $f(Mod(\Gamma)) \subseteq Mod(\phi \wedge \psi)$. But this is clear as $Mod(\phi \wedge \psi) = Mod(\phi) \cap Mod(\psi)$.

2. Say $f(X) \subseteq X$ for any X, and $\phi \in \Gamma$. We must show $\Gamma \mid\!\sim \phi$, or that $f(Mod(\Gamma)) \subseteq Mod(\phi)$. Since $\phi \in \Gamma$, this means that $Mod(\Gamma) \subseteq Mod(\phi)$. And since $f(Mod(\Gamma)) \subseteq Mod(\Gamma)$, we are done.

3. Contraction: clearly $f(X) \subseteq X$ by definition in (A.1).

   Coherence: Now assume $X \subseteq Y$, and say $z \in X \cap f(Y)$. We have to show that $z \in f(X)$. Since $z \in f(Y)$, this means $(\forall y \in Y)[R(z, y)]$. Since $X \subseteq Y$, we can weaken this to conclude that $(\forall y \in X)[R(z, y)]$. But this fact combined with the fact that $z \in X$ means $z \in f(X)$, so we are done.

   Left disjunction: say $\alpha \mid\!\sim \phi \wedge \beta \mid\!\sim \phi$. This means

$$\begin{aligned} & f(Mod(\alpha)) \subseteq Mod(\phi) \\ & f(Mod(\beta)) \subseteq Mod(\phi) \end{aligned} \tag{A.5}$$

   We need to show that $f(Mod(\alpha \vee \beta)) \subseteq Mod(\phi)$.

   It is enough to show that $f(Mod(\alpha \vee \beta)) \subseteq f(Mod(\alpha)) \cup f(Mod(\beta))$. Recall that $Mod(\alpha \vee \beta) = Mod(\alpha) \cup Mod(\beta)$.

Let $x \in f(Mod(\alpha) \cup Mod(\beta))$, and go ahead and assume that $x \notin f(Mod(\alpha))$, and try to show that $x \in f(Mod(\beta))$. $x \in f(Mod(\alpha) \cup Mod(\beta))$ means that $x \in Mod(\alpha) \cup Mod(\beta)$ and $(\forall y \in Mod(\alpha) \cup Mod(\beta))[R(x,y)]$, but either $x \notin Mod(\alpha)$ or $\neg(\forall y \in Mod(\alpha))[R(x,y)]$. Let us split these up into cases.

Say $x \notin Mod(\alpha)$. This means $x \in Mod(\beta)$. Furthermore, $(\forall y \in Mod(\alpha) \cup Mod(\beta))[R(x,y)]$. Hence $(\forall y \in Mod(\beta))[R(x,y)]$, and by definition of f, $x \in f(Mod(\beta))$.

Now on the other hand suppose $\neg(\forall y \in Mod(\alpha))[R(x,y)]$. This contradicts the fact that $(\forall y \in Mod(\alpha) \cup Mod(\beta))[R(x,y)]$ so this case is impossible.

4. Cautious monotonicity: assume $\Gamma \mid\!\sim \phi \wedge \Gamma \mid\!\sim \psi$. This means:

$$\begin{aligned} f(Mod(\Gamma)) &\subseteq Mod(\phi) \\ f(Mod(\Gamma)) &\subseteq Mod(\psi), \end{aligned} \tag{A.6}$$

We need to show that $\Gamma \cup \phi \mid\!\sim \psi$, or $f(Mod(\Gamma \cup \phi)) \subseteq Mod(\psi)$. By (A.6), it is enough to show that $f(Mod(\Gamma \cup \phi)) \subseteq f(Mod(\Gamma))$.

Let $x \in f(Mod(\Gamma \cup \phi))$. This means $x \in Mod(\Gamma \cup \phi)$ and $(\forall y \in Mod(\Gamma \cup \phi))[\neg y < x]$. Hence $x \in Mod(\Gamma)$. All we have to show is that $(\forall y \in Mod(\Gamma))[\neg y < x]$. Say instead there was a $y \in Mod(\Gamma)$ such that $y < x$. Let z be the $<$-minimal element in $Mod(\Gamma)$ such that $z \leq y < x$. Since x is $<$-minimal in $Mod(\Gamma \cup \phi)$, $z \notin Mod(\Gamma \cup \phi)$, which means $z \notin Mod(\phi)$. Hence $z \notin f(Mod(\Gamma))$, which means $(\exists w \in Mod(\Gamma))[w < z]$, a contradiction to z being $<$-minimal. Hence there is no y, and $x \in f(Mod(\Gamma))$.

□

A.2 Proof of Proposition 5.3.2

Proposition 5.3.2 [Properties of f defined by $\mid\!\sim$]. *Let* $\Gamma \mid\!\sim \phi \iff f(Mod(\Gamma)) \subseteq Mod(\phi)$. *Then:*

1. Say $\mid\!\sim$ obeys Inclusion. Then f obeys contraction on the definable subsets of $\mathcal{M}$.

Proof. 1. Say for any Γ, $\Gamma \subseteq \mathcal{C}(\Gamma) = Th(f(Mod(\Gamma)))$. For the purposes of contradiction, assume that in fact $f(X) \nsubseteq X$, for some X. So there is some X and z, such that $z \in f(X) \wedge z \notin X$. Since f is restricted to only definable subsets of $\mathcal{M}$, this means that there is a $B \subseteq \mathcal{L}$ such that $X = Mod(B)$.

By inclusion, we know that $B \subseteq Th(f(Mod(B)))$, or $f(Mod(B)) \subseteq Mod(B)$. But this contradicts our $z \in f(Mod(B)) \wedge z \notin Mod(B)$. Hence contraction of f must hold.

□

A.3 Proof of Proposition 5.4.1

Proposition 5.4.1 [Supraclassicality]. *Let $\langle \mathcal{L}, \vdash, \mid\!\sim, \Gamma \rangle$ be an extended axiomatic formal system with choice. Then the relations $\vdash$ and $\mid\!\sim$ obey supraclassicality.*

Proof.

$$\begin{aligned} \Gamma \vdash \phi &\iff \phi \in Th(Mod(\Gamma)) \\ &\iff (\forall m \in Mod(\Gamma))[m \models \phi] \\ &\implies (\forall m \in f(Mod(\Gamma)))[m \models \phi] \\ &\iff \phi \in Th(f(Mod(\Gamma))) \\ &\iff \Gamma \mid\!\sim \phi \end{aligned} \tag{A.7}$$

The implication in (A.7) is possible because of the contraction of f. □

A.4 Proof of Theorem 5.6.1

Theorem 5.6.1 [Full Retraction Definition is Correct]. *Let $\langle \mathcal{L}, \vdash, \mid\!\sim, \Gamma \rangle$ be an extended axiomatic formal system. Let $\psi_1, \ldots, \psi_n$ be any sequence of formulas from $\Psi_{\mathcal{E}}$. Then*

$$fully\text{-}retractable(\alpha, \langle \mathcal{L}, \vdash, \mathrel{|\!\sim}, \Gamma\rangle) \implies \\ fully\text{-}retractable(\alpha, \langle \mathcal{L}, \vdash, \mathrel{|\!\sim}, \psi_n \cup \ldots \cup \psi_1 \cup \Gamma\rangle). \tag{A.8}$$

Proof. Say $fully\text{-}retractable(\alpha, \langle \mathcal{L}, \vdash, \mathrel{|\!\sim}, \Gamma\rangle)$ holds. This means that:

$$(\forall \Psi \in \Psi_{\mathcal{E}})[(\Psi \cup \Gamma \nvdash \alpha) \wedge (\Psi \cup \Gamma \nvdash \neg\alpha)] \implies \\ (\exists \psi_\emptyset \in \Psi_{\mathcal{E}})[(\psi_\emptyset \cup \Psi \cup \Gamma \mathrel{|\!\not\sim} \alpha) \wedge (\psi_\emptyset \cup \Psi \cup \Gamma \mathrel{|\!\not\sim} \neg\alpha)] \tag{A.9}$$

Call $\Psi_{seq} = \psi_n \cup \ldots \cup \psi_1$. Let $\Psi_0 \in \Psi_{\mathcal{E}}$ and assume that $\Psi_0 \cup \Psi_{seq} \cup \Gamma \nvdash \alpha$ and $\Psi_0 \cup \Psi_{seq} \cup \Gamma \nvdash \neg\alpha$. We must show that there exists a $\psi_\emptyset$ such that $\psi_\emptyset \cup \Psi_0 \cup \Psi_{seq} \cup \Gamma \mathrel{|\!\not\sim} \alpha$ and $\psi_\emptyset \cup \Psi_0 \cup \Psi_{seq} \cup \Gamma \mathrel{|\!\not\sim} \neg\alpha$.

Now, we can substitute $\Psi_0 \wedge \Psi_{seq}$ for Ψ in (A.9) to get our conclusion. □

A.5 Proof of Theorem 5.6.2

Theorem 5.6.2 [Full Addition Definition is Correct]. *Let $\langle \mathcal{L}, \vdash, \mathrel{|\!\sim}, \Gamma\rangle$ be an extended axiomatic formal system. Let $\psi_1, \ldots, \psi_n$ be any sequence of formulas from $\Psi_{\mathcal{E}}$. Then*

$$fully\text{-}addable(\alpha, \langle \mathcal{L}, \vdash, \mathrel{|\!\sim}, \Gamma\rangle) \implies \\ fully\text{-}addable(\alpha, \langle \mathcal{L}, \vdash, \mathrel{|\!\sim}, \psi_n \cup \ldots \cup \psi_1 \cup \Gamma\rangle). \tag{A.10}$$

Proof. Say $fully\text{-}addable(\alpha, \langle \mathcal{L}, \vdash, \mathrel{|\!\sim}, \Gamma\rangle)$ holds. This means that:

$$\begin{aligned}&(\forall \Psi \in \Psi_{\mathcal{E}})[Consis(\alpha, \Psi \cup \Gamma) \implies \\ &\quad(\exists \psi_+ \in \Psi_{\mathcal{E}})[Consis(\psi_+, \Psi \cup \Gamma) \wedge \\ &\qquad (\psi_+ \cup \Psi \cup \Gamma \mathrel{|\!\sim} \alpha) \wedge \\ &\qquad fully\text{-}retractable(\alpha, \langle \mathcal{L}, \vdash, \mathrel{|\!\sim}, \psi_+ \cup \Psi \cup \Gamma\rangle) \wedge \\ &\qquad Consis(\neg\alpha, \Psi \cup \Gamma) \implies Consis(\neg\alpha, \psi_+ \cup \Psi \cup \Gamma)]]\end{aligned} \tag{A.11}$$

Call $\Psi_{seq} = \psi_n \cup \ldots \cup \psi_1$. Let $\Psi_0 \in \Psi_{\mathcal{E}}$ and assume that

$Consis(\alpha, \Psi_0 \cup \Psi_{seq} \cup \Gamma)$. We must produce a ψ_+ such that

1. $Consis(\psi_+, \Psi_0 \cup \Psi_{seq} \cup \Gamma)$

2. $\psi_+ \cup \Psi_0 \cup \Psi_{seq} \cup \Gamma \mid\sim \alpha$, and

3. $fully\text{-}retractable(\alpha, \langle \mathcal{L},\ \vdash,\ \mid\sim,\ \psi_+ \cup \Psi_0 \cup \Psi_{seq} \cup \Gamma\rangle)$

4. $Consis(\neg\alpha, \Psi_0 \cup \Psi_{seq} \cup \Gamma) \implies Consis(\neg\alpha, \psi_+ \cup \Psi_0 \cup \Psi_{seq} \cup \Gamma)]]$

Substitute $\Psi_0 \wedge \Psi_{seq}$ in (A.11) to get our result. □

A.6 Proof of Theorem 5.6.3

Theorem 5.6.3 [Full Retraction and Addition Require Infinite Languages]. *Let $\langle \mathcal{L},\ \vdash,\ \mid\sim,\ \Gamma\rangle$ be an extended axiomatic formal system with choice which obeys faithfulness and left logical equivalence, and say there is a non-empty subset $\mathcal{L}^* \subseteq \mathcal{L}$, which is fully retractable and fully addable in Γ, and not trivially so, for either case.*

Then $\mathcal{L}$ must have infinitely many formulas.

Proof. Pick some $\alpha \in \mathcal{L}^*$. Since it's not trivially fully retractable, we know for every $\Psi \in \Psi_{\mathcal{E}}$, $\Psi \cup \Gamma \not\vdash \alpha \wedge \Psi \cup \Gamma \not\vdash \neg\alpha$. Let Ψ be $\top$.

By full retraction there is a formula $\psi_\emptyset(\alpha, \Gamma)$ such that

$$(\psi_\emptyset(\alpha, \Gamma) \cup \Gamma \not\mid\sim \alpha) \wedge (\psi_\emptyset(\alpha, \Gamma) \cup \Gamma \not\mid\sim \neg\alpha) \tag{A.12}$$

For simplicity, let us abbreviate $\psi_\emptyset(\alpha, \Gamma) \cup \Gamma$ as $\Gamma^0_\emptyset$. Supraclassicality applied to (A.12) results in the inference

$$\Gamma^0_\emptyset \not\vdash \alpha \wedge \Gamma^0_\emptyset \not\vdash \neg\alpha \equiv \quad Consis(\neg\alpha, \Gamma^0_\emptyset) \wedge Consis(\alpha, \Gamma^0_\emptyset) \tag{A.13}$$

Now by full addition ($\psi_\emptyset(\alpha, \Gamma) \in \Psi_{\mathcal{E}}$), we know there is a $\psi_+(\alpha, \Gamma^0_\emptyset)$ such that

$$\begin{aligned}
&Consis(\psi_+(\alpha, \Gamma^0_\emptyset), \Gamma^0_\emptyset) \wedge \\
&\psi_+(\alpha, \Gamma^0_\emptyset) \cup \Gamma^0_\emptyset \mathrel{|\!\sim} \alpha \wedge \\
&\textit{fully-retractable}(\alpha, \langle \mathcal{L},\ \vdash,\ \mathrel{|\!\sim},\ \psi_+(\alpha, \Gamma^0_\emptyset) \cup \Gamma^0_\emptyset \rangle) \wedge \\
&\Gamma^0_\emptyset \nvdash \alpha \implies \psi_+(\alpha, \Gamma^0_\emptyset) \cup \Gamma^0_\emptyset \nvdash \alpha
\end{aligned} \tag{A.14}$$

From (A.13) and the last implication of (A.14), we can infer that $\psi_+(\alpha, \Gamma^0_\emptyset) \cup \Gamma^0_\emptyset \nvdash \alpha$.

We can also show that $\psi_+(\alpha, \Gamma^0_\emptyset) \cup \Gamma^0_\emptyset \nvdash \neg\alpha$ – if it did, then $\psi_+(\alpha, \Gamma^0_\emptyset) \cup \Gamma^0_\emptyset \mathrel{|\!\sim} \neg\alpha$ by supraclassicality. By Proposition 5.3.1, $\mathrel{|\!\sim}$ satisfies right conjunction, so $\psi_+(\alpha, \Gamma^0_\emptyset) \cup \Gamma^0_\emptyset \mathrel{|\!\sim} \alpha \wedge \neg\alpha$. Unraveling the definition of $\mathrel{|\!\sim}$ in terms of choice functions, we see that $f(Mod(\psi_+(\alpha, \Gamma^0_\emptyset) \cup \Gamma^0_\emptyset)) \subseteq Mod(\alpha \wedge \neg\alpha) = \emptyset$. Since f obeys faithfulness, $Mod(\psi_+(\alpha, \Gamma^0_\emptyset) \cup \Gamma^0_\emptyset) = \emptyset$. Hence this means that $\neg(\exists m \in \mathcal{M})[m \models \psi_+(\alpha, \Gamma^0_\emptyset) \cup \Gamma^0_\emptyset]$.

From (A.14) $Consis(\psi_+(\alpha, \Gamma^0_\emptyset), \Gamma^0_\emptyset)$, which means $(\exists m \in \mathcal{M})[m \models \Gamma_\emptyset \wedge m \not\models \neg\psi_+(\alpha, \Gamma^0_\emptyset)] \equiv (\exists m \in \mathcal{M})[m \models \psi_+(\alpha, \Gamma^0_\emptyset) \cup \Gamma^0_\emptyset]$, a direct contradiction.

Let us rename $\psi_+(\alpha, \Gamma^0_\emptyset) \cup \Gamma^0_\emptyset = \Gamma^0_+$. Since $\Gamma^0_+ \nvdash \alpha$ and $\Gamma^0_+ \nvdash \neg\alpha$, we can apply full retraction again (as $\Gamma^0_+ = \psi_+(\alpha, \Gamma^0_\emptyset) \cup \psi_\emptyset(\alpha, \Gamma) \cup \Gamma$ and $\psi_+(\alpha, \Gamma^0_\emptyset) \wedge \psi_\emptyset(\alpha, \Gamma) \in \Psi_{\mathcal{E}}$). This results in

$$\psi_\emptyset(\alpha, \Gamma^0_+) \cup \Gamma^0_+ \mathrel{|\!\not\sim} \alpha \wedge \psi_\emptyset(\alpha, \Gamma^0_+) \cup \Gamma^0_+ \mathrel{|\!\not\sim} \neg\alpha. \tag{A.15}$$

We can see that this can proceed indefinitely. Define inductively

$$\begin{aligned}
&\Gamma^0_\emptyset \equiv_{def} \psi_\emptyset(\alpha, \Gamma) \cup \Gamma \\
&\Gamma^0_+ \equiv_{def} \psi_+(\alpha, \Gamma^0_\emptyset) \cup \Gamma^0_\emptyset \\
&\Gamma^{i+1}_\emptyset \equiv_{def} \psi_\emptyset(\alpha, \Gamma^i_+) \cup \Gamma^i_+ \\
&\Gamma^{i+1}_+ \equiv_{def} \psi_+(\alpha, \Gamma^{i+1}_\emptyset) \cup \Gamma^{i+1}_\emptyset \\
&\Gamma^{i+1}_\emptyset \mathrel{|\!\not\sim} \alpha \wedge \Gamma^{i+1}_\emptyset \mathrel{|\!\not\sim} \neg\alpha \\
&\Gamma^{i+1}_+ \mathrel{|\!\sim} \alpha
\end{aligned} \tag{A.16}$$

Hence $\Gamma^i_\emptyset = \psi^{i-1}_\emptyset, \psi^{i-1}_+, \ldots, \psi^0_\emptyset, \psi^0_+, \psi_\emptyset(\alpha, \Gamma), \Gamma$ while $\Gamma^i_+ = \psi^i_+, \psi^{i-1}_\emptyset, \psi^{i-1}_+, \ldots, \psi^0_\emptyset, \psi^0_+, \psi_\emptyset(\alpha, \Gamma), \Gamma$, where we abbreviate $\psi^j_\emptyset = \psi_\emptyset(\alpha, \Gamma^j_+)$ and

$\psi_+^j = \psi_+(\alpha, \Gamma_\emptyset^j)$.

Now instead assume that our $\mathcal{L}$ has finitely many formulas. This means that there are finitely many formulas of the form $\psi_\emptyset^j$, call this number $n+1$. Now consider what happens when we have

$$\begin{aligned}&\Gamma_+^n \mid\!\sim \alpha, \ \textit{where}\\ &\Gamma_+^n = \psi_+^n \cup \psi_\emptyset^{n-1} \cup \psi_+^{n-1} \cup \ \ldots \psi_\emptyset^0 \cup \psi_+^0 \cup \psi_\emptyset(\alpha, \Gamma) \cup \Gamma\end{aligned} \tag{A.17}$$

By our inductive argument, $\psi_\emptyset(\alpha, \Gamma_+^n) \cup \Gamma_+^n \not\mid\!\sim \alpha$, but $\psi_\emptyset(\alpha, \Gamma_+^n) \cup \Gamma_+^n = \psi_\emptyset^n, \psi_+^n, \psi_\emptyset^{n-1}, \psi_+^{n-1}, \ldots \psi_\emptyset^0, \psi_+^0, \psi_\emptyset(\alpha, \Gamma), \Gamma$, which is logically equivalent to $\psi_+^n, \psi_\emptyset^{n-1}, \psi_+^{n-1}, \ldots \psi_\emptyset^0, \psi_+^0, \psi_\emptyset(\alpha, \Gamma), \Gamma$, since there are at most $n+1$ different kinds of $\psi_\emptyset$. But by definition again, $\psi_+^n, \psi_\emptyset^{n-1}, \psi_+^{n-1}, \ldots \psi_\emptyset^0, \psi_+^0, \psi_\emptyset(\alpha, \Gamma), \Gamma = \Gamma_+^n$ and $\Gamma_+^n \mid\!\sim \alpha$, a contradiction. □

A.7 Proof of Corollary 5.6.1

Corollary 5.6.1 [Nonmonotonicity of $\mid\!\sim$]. $\mid\!\sim$*, for the conditions in Theorem 5.6.3 must be nonmonotonic.*

Proof. This is by inspection of the workings of Γ_+^n and $\Gamma_\emptyset^n$, where $\Gamma_\emptyset^{n+1} = \psi_\emptyset^n \cup \Gamma_+^n$, and while $\Gamma_+^n \mid\!\sim \alpha$, $\Gamma_\emptyset^{n+1} \not\mid\!\sim \alpha$. □

A.8 Proof of Proposition 6.2.1

Proposition 6.2.1 [Consequences of f^* and g^*]. *Let f^* and g^* be as in definition 6.2.3. then:*

1. *f^* and g^* satisfy contraction.*

2. *If f satisfies coherence, so does f^*. The same goes for g^*.*

3. *f^* satisfies* right interchangeability, *defined as:*

$$(a, b) \in f^*(w) \wedge (a, c) \in w \implies (a, c) \in f^*(w). \tag{A.18}$$

g^* satisfies left interchangeability.

4. *If f satisfies faithfulness, so does f^*.*

5. *If f satisfies ϕ-reflection for $\phi \in \mathcal{L}_1$, so does f^*.*

Proof. 1. Contraction: $f^*(W) \subseteq W$ trivially by definition.

2. Right Interchangeability: Say $(a, b) \in f^*(W) \wedge (a, c) \in W$. The first conjunct allows us to infer that $a \in f(\{s \mid (s, t) \in W\})$. And to show that $(a, c) \in f^*(W)$ we only need $(a, c) \in W \wedge a \in f(\{s \mid (s, t) \in W\})$, which are given.

   The proof of Left Interchangeability for g^* is symmetric.

3. Coherence. We must show that if $X \subseteq Y$, then $X \cap f^*(Y) \subseteq f^*(X)$. So say $X \subseteq Y$ and $(a, b) \in X, f^*(Y)$. We must show that $(a, b) \in f^*(X)$, or that $(a, b) \in X$ and that $a \in f(\{s \mid (s, t) \in X\})$. The first is given. We just need to show that $a \in f(\{s \mid (s, t) \in X\})$.

   Since $X \subseteq Y$, $\{s \mid (s, t) \in X\} \subseteq \{s \mid (s, t) \in Y\}$. Since f satisfies coherence, this means that $\{s \mid (s, t) \in X\} \cap f(\{s \mid (s, t) \in Y\}) \subseteq f(\{s \mid (s, t) \in X\})$. So it is enough to show that $a \in \{s \mid (s, t) \in X\}$ and $a \in f(\{s \mid (s, t) \in Y\})$. But this holds, as the first is given, and as $(a, b) \in f^*(Y)$, this means $a \in f(\{s \mid (s, t) \in Y\})$.

4. Faithfulness: Say $f^*(X) = \emptyset$. This means that

$$\{(m_1, m_2) \in X \mid m_1 \in f([X]_\ell)\} = \emptyset. \tag{A.19}$$

   This is either because X is empty, in which case we have our conclusion, or $f([X]_\ell) = \emptyset$. By faithfulness of f, this second case is only possible if $[X]_\ell$ is empty, which is only true when $X = \emptyset$.

5. ϕ-reflection: Let ϕ be an $\mathcal{L}_1$-formula. Say $\phi, \neg\phi \notin Th(f^*(X)) \wedge \phi, \neg\phi \notin Th(Y) \wedge Y \subseteq X$. We must show that $\phi, \neg\phi \notin Th(f^*(Y))$.

The assumptions means there are models $(m_1^+, m_2^+), (m_1^-, m_2^-) \in f^*(X)$ such that $m_1^+ \models \phi$ and $m_1^- \models \neg\phi$. By definition of f^*, $m_1^+, m_1^- \in f([X]_\ell)$. There are also $(n_1^+, n_2^+), (n_1^-, n_2^-) \in Y$ such that $n_1^+ \models \phi$ and $n_1^- \models \neg\phi$, and $n_1^+, n_1^- \in [Y]_\ell$. Since $Y \subseteq X$, $[Y]_\ell \subseteq [X]_\ell$. From reflection on f we then have that there are models $p_1^+, p_1^- \in f([Y]_\ell)$.

Since f satisfies coherence, $p_1^+, p_1^- \in [Y]_\ell$, so that there are p_2^+, p_2^- such that $(p_1^+, p_2^+), (p_1^-, p_2^-) \in Y$. But then this means $(p_1^+, p_2^+), (p_1^-, p_2^-) \in f^*(Y)$.

□

A.9 Proof of the Upwardly Free *Ab* Lemma

Upwardly Free *Ab* Lemma. *Let Φ be any $\mathcal{L}_1$-theory and $\Psi \in Elab^+(\mathcal{L}_1, \mathcal{L}_{Ab})$. Say we have an $\mathcal{L}_{1,Ab}$-structure (m_1, m_2) such that $(m_1, m_2) \models_{1,Ab} \Phi \cup \Psi \cup \Gamma_{Ab}$, and call ℓ^{max} the $>$-highest symbol of Param mentioned in Ψ.*

Call $\Lambda^{max} = \{\ell \in Param \mid \ell > \ell^{max} \in \Gamma_{Ab}\}$. Let Λ be a arbitrary subset of Λ^{max}. Then there is a model $(m_1, m_2^*) \models_{1,Ab} \Phi \cup \Psi \cup \Gamma_{Ab}$ where m_2^* looks just like m_2 except that $Ab^{m_2^*} = (Ab^{m_2} \setminus \Lambda^{max}) \cup \Lambda$. In other words, we can assign the remaining parameters $>$-than those mentioned in Ψ arbitrarily to m_2's extension of Ab without affecting its satisfaction of $\Phi \cup \Psi \cup \Gamma_{Ab}$.

Proof. Let (m_1, m_2) be the model of $\Phi \cup \Psi \cup \Gamma_{Ab}$. Construct a copy (m_1, m_2^*) of (m_1, m_2) except that $Ab^{m_2^*} = (Ab^{m_2} \setminus \Lambda^{max}) \cup \Lambda$. We must show that $(m_1, m_2^*) \models_{1,Ab} \Phi \cup \Psi \cup \Gamma_{Ab}$, or:

$$(m_1, m_2^*) \models_{1,Ab} \Phi \cup \Psi \cup \Gamma_{Ab} \iff m_1 \models_1 \Phi \wedge (m_1, m_2^*) \models_{1,Ab} \Psi \ \wedge m_2^* \models_{Ab} \Gamma_{Ab} \tag{A.20}$$

$m_1 \models_1 \Phi$ as that is given. Similarly $m_2^* \models_{Ab} \Gamma_{Ab}$, as *Ab* is not mentioned in Γ_{Ab}, and that is the only way m_2^* differs from m_2.

Now we must show $(m_1, m_2^*) \models_{1,Ab} \Psi$. There are three cases, one for each of the possible forms of statement in Ψ.

Say $\phi \in \mathcal{L}_1$ is a member of Ψ. Since $(m_1, m_2) \models_{1,Ab} \phi$, $(m_1, m_2^*) \models_{1,Ab} \phi$, as this only depends on m_1.

Say $Ab(\ell_i) \vee \phi_i$ is a member of Ψ. Assume $(m_1, m_2^*) \models_{1,Ab} \neg\phi_i$ and show $(m_1, m_2^*) \models_{1,Ab} Ab(\ell_i)$: $(m_1, m_2^*) \models_{1,Ab} \neg\phi_i$ means $m_1 \models_1 \neg\phi_i$, so $(m_1, m_2) \models_{1,Ab} \neg\phi_i$. $(m_1, m_2) \models_{1,Ab} \Psi$, so $(m_1, m_2) \models_{1,Ab} Ab(\ell_i)$. Since ℓ_i is mentioned in Ψ and ℓ^{max} is the $>$-highest parameter mentioned in Ψ, $\neg\ell_i > \ell^{max}$, so $\ell_i \notin \Lambda^{max}$, so $\ell_i \in Ab^{m_2^*}$, and therefore $(m_1, m_2^*) \models_{1,Ab} Ab(\ell_i)$.

Now consider statements of the form $(\forall_{Ab}\ x)[Ab(\ell_j) \wedge x < \ell_j \implies Ab(x)]$. Let us assume that $(m_1, m_2^*) \models_{1,Ab} Ab(\ell_j) \wedge a < \ell_j$ for some $a \in |m_2^*|$. We must show $(m_1, m_2^*) \models_{1,Ab} Ab(a)$: Again, $\neg\ell_j > \ell^{max}$, so $\ell_j \notin \Lambda^{max}$ and therefore $\ell_j \notin \Lambda$. Since $(m_1, m_2^*) \models_{1,Ab} Ab(\ell_j)$ this means $\ell_j \in Ab^{m_2^*}$, so that $\ell_j \in Ab^{m_2}$. Hence $(m_1, m_2) \models_{1,Ab} Ab(\ell_j)$. Furthermore, since m_2 and m_2^* do not differ on $<$, $(m_1, m_2) \models_{1,Ab} a < \ell_j$, from which we can infer $(m_1, m_2) \models Ab(a)$.

Now $a < \ell_j \leq \ell^{max}$, so a cannot possibly be mentioned in Λ^{max}. But then $a \in Ab^{m_2^*}$, as have just shown that $a \in Ab^{m_2}$. So $(m_1, m_2^*) \models_{1,Ab} \Psi$ as well, and we have our model. □

A.10 Proof of the Downwardly Free *Ab* Lemma

Downwardly Free *Ab* Lemma. *Let Φ be any $\mathcal{L}_1$-theory and $\Psi \in Elab^+(\mathcal{L}_1, \mathcal{L}_{Ab})$. Say we have an $\mathcal{L}_{1,Ab}$-structure (m_1, m_2) such that $(m_1, m_2) \models_{1,Ab} \Phi \cup \Psi \cup \Gamma_{Ab}$. Call ℓ^{max} the $>$-highest symbol of Param mentioned in Ψ.*

Let n_2 be any $\mathcal{L}_{Ab}$-model of Γ_{Ab}, such that $\{a \in |m_2| \mid a \leq \ell^{max}\} \subseteq Ab^{n_2}$. (Note that this leaves n_2 unspecified for elements $>$-than ℓ^{max}.) Then $(m_1, n_2) \models_{1,Ab} \Phi \cup \Psi \cup \Gamma_{Ab}$.

Proof. Clearly, $(m_1, n_2) \models_{1,Ab} \Phi \cup \Gamma_{Ab}$ by definition. We have to verify it remains true for all versions of Ψ.

If Ψ contains an $\mathcal{L}_1$-formula conjunct ϕ, then $(m_1, n_2) \models_{1,Ab} \phi$ trivially.

If Ψ has a formula of the form $Ab(\ell) \vee \phi$, then by definition, $\ell \leq \ell^{max}$, so $n_2 \models_2 Ab(\ell)$, and $(m_1, n_2) \models_{1,Ab} Ab(\ell) \vee \phi$.

Finally, say Ψ has a formula of the form $(\forall_{Ab}\ x)[Ab(\ell_j) \wedge x < \ell_j \implies Ab(x)]$. Say $(m_1, n_2) \models_{1,Ab} Ab(\ell_j) \wedge a < \ell_j$ for some $a \in |n_2|$. $a < \ell_j \leq \ell^{max}$, so $n_2 \models_{Ab} Ab(a)$ and we are done.

□

A.11 Proof of Lemma 8.3.1

Lemma 8.3.1 [Every $<_{Ab}$ chain ends]. *Let $\Psi \in Elab^+(\mathcal{L}_1, \mathcal{L}_{Ab})$ and Φ $\mathcal{L}_1$-theory. Say $(m_1^0, m_2^0) \models_{1,Ab} \Psi \cup \Phi \cup \Gamma_{Ab}$.*

Then there exists a $(n_1, n_2) \models_{1,Ab} \Psi \cup \Phi \cup \Gamma_{Ab}$ which is Ab-minimal amongst $Mod_{1,Ab}(\Psi \cup \Phi \cup \Gamma_{Ab})$ and $n_2 \leq_{Ab} m_2^0$.

Proof. Call $\Delta = \Psi \cup \Phi \cup \Gamma_{Ab}$. We must show that $Mod_{1,Ab}(\Delta)$ has an *Ab*-minimal element:

$$(\exists(n_1, n_2) \in Mod_{1,Ab}(\Delta))(\forall(b_1, b_2) \in Mod_{1,Ab}(\Delta))[\neg b_2 <_{Ab} n_2] \tag{A.21}$$

Let us say instead that it does not, that in fact:

$$(\forall(n_1, n_2) \in Mod(\Delta))(\exists(b_1, b_2) \in Mod(\Delta))[b_2 <_{Ab} n_2] \tag{A.22}$$

Let X be the set of all models $(a_1, a_2) \models \Delta$ such that $|a_2| = |m_2^0|, <^{a_2} = <^{m_2^0}$, $\ell^{a_2} = \ell^{m_2^0}$ for all $\ell \in Param$ and $Ab^{a_2} \subseteq Ab^{m_2^0}$.

Now, assume that:

$$\Psi = \phi \wedge \bigwedge_{1 \leq i \leq p} Ab(\ell_i) \vee \phi_i \wedge \bigwedge_{p+1 \leq j \leq r} (\forall_{Ab}\ x)[Ab(\ell_j) \wedge x < \ell_j \implies Ab(x)] \tag{A.23}$$

Over this set of elements $\ell_1, \ldots \ell_r$ mentioned in Ψ, consider the model $(m_1, m_2) \in X$ with the smallest extension of elements of $\ell_1, \ldots \ell_r$ in Ab^{m_2}. (m_1, m_2) satisfies:

$$(\forall (a_1, a_2) \in X)[\neg(Ab^{a_2} \cap \{\ell_1, \ldots, \ell_r\} \subset Ab^{m_2} \cap \{\ell_1, \ldots, \ell_r\})] \tag{A.24}$$

Now from (m_1, m_2) construct another (m_1, n_2) such that n_2 is the same as m_2 except that

$$\begin{aligned} Ab^{n_2} \quad = \quad & Ab^{m_2} \cap \{\ell_1, \ldots, \ell_r\} \cup \\ & \{a \in |m_2| \mid \ell_j \in Ab^{m_2} \wedge (\forall_{Ab}\ x)[Ab(\ell_j) \wedge x < \ell_j \implies Ab(x)] \in \Psi \wedge \\ & \qquad (a, \ell_j) \in <^{m_2}\} \end{aligned} \tag{A.25}$$

In other words, we assert in Ab^{n_2} precisely those elements which must be true in Ψ and no more. By this construction, $n_2 \leq_{Ab} m_2$.

Now apply (m_1, n_2) to our (A.22). First we show that $(m_1, n_2) \models_{1,Ab} \Delta$. $(m_1, n_2) \models_{1,Ab} \Phi \cup \Gamma_{Ab}$ as those parts of the language were unchanged. Showing $(m_1, n_2) \models_{1,Ab} \Psi$ is a little more complicated:

$(m_1, n_2) \models_{1,Ab} \phi$ as this only depends on m_1 which entails Ψ. If $(m_1, n_2) \models_{1,Ab} \neg\phi_i$, then $(m_1, m_2) \models_{1,Ab} \neg\phi_i$ and $(m_1, m_2) \models_{1,Ab} Ab(\ell_i)$, which means by definition $(m_1, n_2) \models_{1,Ab} Ab(\ell_i)$. As for statements $(\forall_{Ab}\ x)[Ab(\ell_j) \wedge x < \ell_j \implies Ab(x)]$, say $(m_1, n_2) \models_{1,Ab} Ab(\ell_j) \wedge a < \ell_j$ for some $a \in |n_2| = |m_2|$. By definition, $(m_1, m_2) \models_{1,Ab} Ab(\ell_j) \wedge a < \ell_j$, and therefore $(m_1, m_2) \models_{1,Ab} Ab(a)$. From construction of Ab^{n_2} we see that $a \in Ab^{n_2}$.

Since $n_2 \leq_{Ab} m_2$, and $(m_1, n_2) \models_{1,Ab} \Delta$, $(m_1, n_2) \in X$. Now, (A.22) says we can find a $(b_1, b_2) \models_{1,Ab} \Delta$ such that $b_2 <_{Ab} n_2$. This means $b_2 \in X$. Call $a \in Ab^{n_2} \setminus Ab^{b_2}$. Now clearly, a does not correspond to a named element ($\ell^{n_2} \neq a$), because that would contradict (A.24). Hence it must be some unnamed element, and it must be the case that $\ell_j \in Ab^{m_2} \wedge (\forall_{Ab}\ x)[Ab(\ell_j) \wedge x < \ell_j \implies Ab(x)] \in \Psi \wedge (a, \ell_j) \in <^{m_2}$. Now $\ell_j \in Ab^{b_2}$ by (A.24), and $(a, l_j) \in <^{b_2}$ since $b_2 <_{Ab} n_2$. Since $(b_1, b_2) \models_{1,Ab} \Psi$, $(b_1, b_2) \models_{1,Ab} Ab(a)$, but this contradicts the fact that $a \in Ab^{n_2} \setminus Ab^{b_2}$.

Hence there must be *Ab*-minimal elements, and (m_1, n_2) is one. And $n_2 \leq_{Ab} m_2$ by construction, and $m_2 \leq_{Ab} m_2^0$ as they are both in X.

$\square$

A.12 Proof of Corollary 8.3.2

Corollary 8.3.2 [g^* **is Non-Empty**]. *Let* $\Psi \in Elab^+(\mathcal{L}_1, \mathcal{L}_{Ab})$ *and* Φ *a consistent* $\mathcal{L}_1$*-theory. Furthermore, if* Ψ *contains a conjunct with a pure* $\mathcal{L}_1$*-formula* ($\Psi \equiv \phi \wedge \bigwedge_i Ab(\ell_i) \vee \phi_i \wedge \bigwedge_j (\forall_{Ab}\ x)[Ab(\ell_j) \wedge x < \ell_j \implies Ab(x)]$)*,* $\Phi \cup \phi$ *is consistent.*

Then $g^*(Mod(\Psi \cup \Phi \cup \Gamma_{Ab})) \neq \emptyset$*.*

Proof. Call $\Delta = \Psi \cup \Phi \cup \Gamma_{Ab}$.

$$g^*(Mod(\Delta)) \;=\; \{(m_1, m_2) \in Mod(\Delta) \mid (\forall (n_1, n_2) \in Mod(\Delta))[\neg(n_2 <_{Ab} m_2)]\} \tag{A.26}$$

To show that $g^*(\Delta)$ is non-empty, we must show $Mod(\Delta)$ is non-empty, and that there is at least one element $(m_1, m_2) \in Mod(\Delta)$ such that $(\forall (n_1, n_2) \in Mod(\Delta))[\neg(n_2 <_{Ab} m_2)]$.

Showing $Mod(\Delta)$ is non-empty is straightforward. Since $\Phi \cup \phi$ is consistent, let m_1^0 be a model of it. Let m_2^0 be a Herbrand model of Γ_{Ab}, where $|m_2^0| = Param$, $<$ is defined according to Γ_{Ab}. Set $Ab^{m_2^0} = |m_2^0|$. $(m_1^0, m_2^0) \models_{1,Ab} \Delta$ trivially.

For the second part, since we have a model of Δ, we can use Lemma 8.3.1 to produce a witness $(m_1, m_2) \models_{1,Ab} \Delta$ and $(\forall (n_1, n_2) \in Mod(\Delta))[\neg n_2 <_{Ab} m_2]$. $\square$

A.13 Proof of the Monotonic Retraction Lemma

Monotonic Retraction Lemma ($[g^*(Mod(\Delta))]_\ell \subseteq [g^*(Mod(\Delta \cup \psi_\emptyset))]_\ell$**).** *Let* Φ *be some* $\mathcal{L}_1$*-theory,* $\Psi \in Elab^+(\mathcal{L}_1, \mathcal{L}_{Ab})$*, and* Γ_{Ab} *as defined. Say* $\Delta = \Psi \cup \Phi \cup \Gamma_{Ab}$*. Let* α *be a* $\mathcal{L}_1$*-formula and* $\psi_\emptyset = (Ab(\ell^+) \vee \alpha) \wedge (Ab(\ell^-) \vee \neg\alpha)$*, where* $\ell^+, \ell^- \in Param$ *and are* $>$*-than any labels mentioned in* Ψ*.*

Then $[g^*(Mod(\Delta))]_\ell \subseteq [g^*(Mod(\Delta \cup \psi_\emptyset))]_\ell$*.*

Proof. Let $(m_1, m_2) \in g^*(Mod(\Delta))$. We must show that there is some m_2' such that $(m_1, m_2') \in g^*(Mod(\Delta \cup \psi_\emptyset))$. Since $(m_1, m_2) \in g^*(Mod(\Delta))$, this means that:

$$\begin{aligned} &(m_1, m_2) \models_{1,Ab} \Delta \\ &(\forall (n_1, n_2) \models_{1,Ab} \Delta)[\neg n_2 <_{Ab} m_2] \end{aligned} \tag{A.27}$$

From m_2 construct a similar m_2^*, except that

$$Ab^{m_2^*} = \begin{cases} Ab^{m_2} \cup \{\ell^+\} \setminus \{\ell^-\} & m_1 \models_1 \neg\alpha \\ Ab^{m_2} \cup \{\ell^-\} \setminus \{\ell^+\} & m_1 \models_1 \alpha \end{cases}$$

By construction $(m_1, m_2^*) \models_{1,Ab} \psi_\emptyset$. Also, since ℓ^+, ℓ^- are symbols greater than those mentioned in Δ, by the Upwardly Free *Ab* Lemma, $(m_1, m_2^*) \models_{1,Ab} \Delta$. To show that $(m_1, m_2^*) \in g^*(Mod(\Delta \cup \psi_\emptyset))$, all we have to show is that

$$(\forall (n_1, n_2) \models_{1,Ab} \Delta \cup \psi_\emptyset)[\neg n_2 <_{Ab} m_2^*] \tag{A.28}$$

Let $(n_1, n_2) \models_{1,Ab} \Delta \cup \psi_\emptyset$, and for the purposes of contradiction, assume that in fact $n_2 <_{Ab} m_2^*$. This means that n_2 looks just like m_2^*, except that there is some element $\ell \in Ab^{m_2^*}$ and $\ell \notin Ab^{n_2}$. This element cannot be ℓ^+ or ℓ^-, because since only one of them appears in $Ab^{m_2^*}$, if it was removed, then it would be impossible for $n_2 \models_{1,Ab} \psi_\emptyset$. So it must be some other element. Hence we can say that

$$Ab^{n_2} \setminus \{\ell^+, \ell^-\} \subset Ab^{m_2^*} \setminus \{\ell^+, \ell^-\} \subseteq Ab^{m_2}. \tag{A.29}$$

Now consider the structure n_2^* which is just like n_2, except that $Ab^{n_2^*} = Ab^{n_2} \setminus \{\ell^+, \ell^-\}$. By the Upwardly Free *Ab* Lemma, $(n_1, n_2^*) \models_{1,Ab} \Delta$. Also, we see by construction that $n_2^* <_{Ab} m_2$. But this fact contradicts (A.27) that m_2 is *Ab*-minimal over this set. So it must be the case that $(m_1, m_2^*) \in g^*(Mod(\Delta \cup \psi_\emptyset))$, and therefore that $[g^*(Mod(\Delta))]_\ell \subseteq [g^*(Mod(\Delta \cup \psi_\emptyset))]_\ell$. □

A.14 Proof of the Full Retractability Lemma

.

Full Retractability Lemma. *Let* $\Psi_{\mathcal{E}} = Elab^{+}(\mathcal{L}_1, \mathcal{L}_{Ab})$, τ *an element of* $Elab^{+}(\mathcal{L}_1, \mathcal{L}_{Ab})$, Φ *a consistent* $\mathcal{L}_1$*-theory, and* α *any* $\mathcal{L}_1$*-formula. Say* f*, the choice function underlying* S_1*, satisfies contraction, coherence, faithfulness, and* α*-reflection.*

Then $fully\text{-}retractable(\alpha, \langle \mathcal{L}_{1,Ab},\ \vdash_{1,Ab},\ \mid\!\sim_{1,Ab},\ \tau \cup \Phi \cup \Gamma_{Ab}\rangle)$.

Proof. We need to show that

$$\begin{aligned}(\forall \Psi \in \Psi_{\mathcal{E}})[\ \ & (\Psi \cup \tau \cup \Phi \cup \Gamma_{Ab} \not\vdash_{1,Ab} \alpha) \wedge \\ & (\Psi \cup \tau \cup \Phi \cup \Gamma_{Ab} \not\vdash_{1,Ab} \neg\alpha)] \implies \\ (\exists \psi_{\emptyset} \in \Psi_{\mathcal{E}})[\ \ & (\psi_{\emptyset} \cup \Psi \cup \tau \cup \Phi \cup \Gamma_{Ab} \not\mid\!\sim_{1,Ab} \alpha) \wedge \\ & (\psi_{\emptyset} \cup \Psi \cup \tau \cup \Phi \cup \Gamma_{Ab} \not\mid\!\sim_{1,Ab} \neg\alpha)]\end{aligned} \tag{A.30}$$

Let Ψ be any member of $Elab^{+}(\mathcal{L}_1, \mathcal{L}_{Ab})$, and call $\Delta = \Psi \cup \tau \cup \Phi \cup \Gamma_{Ab}$. Let $\ell^{+}, \ell^{-} \in Param$ be two fresh symbols not mentioned in $\Psi \cup \tau$, which are $>$ than every other $Param$ symbol mentioned. (This is possible as both are finite formulas.) Take $\psi_{\emptyset}$ as:

$$\psi_{\emptyset} \quad = \quad (Ab(\ell^{+}) \vee \alpha) \wedge (Ab(\ell^{-}) \vee \neg\alpha) \tag{A.31}$$

Now we can rewrite (A.30) as:

$$\begin{aligned}&[f^{*}(Mod(\Delta))]_{\ell} \not\subseteq Mod(\alpha) \wedge [f^{*}(Mod(\Delta))]_{\ell} \not\subseteq Mod(\neg\alpha) \implies \\ &\qquad [f^{*}(g^{*}(Mod(\Delta \cup \psi_{\emptyset})))]_{\ell} \not\subseteq Mod(\alpha) \wedge \\ &\qquad [f^{*}(g^{*}(Mod(\Delta \cup \psi_{\emptyset})))]_{\ell} \not\subseteq Mod(\neg\alpha)\end{aligned} \tag{A.32}$$

Assume $[f^{*}(Mod(\Delta))]_{\ell} \not\subseteq Mod(\alpha)$ and $[f^{*}(Mod(\Delta))]_{\ell} \not\subseteq Mod(\neg\alpha)$. We must show that $[f^{*}(g^{*}(Mod(\Delta \cup \psi_{\emptyset})))]_{\ell} \not\subseteq Mod(\alpha) \wedge [f^{*}(g^{*}(Mod(\Delta \cup \psi_{\emptyset})))]_{\ell} \not\subseteq Mod(\neg\alpha)$.

By α-reflection, it is enough to show that $g^{*}(Mod(\Delta \cup \psi_{\emptyset})) \subseteq Mod(\Delta)$, and that $g^{*}(Mod(\Delta \cup \psi_{\emptyset})) \not\subseteq Mod(\alpha)$ nor $g^{*}(Mod(\Delta \cup \psi_{\emptyset})) \not\subseteq Mod(\neg\alpha)$.

This first is easily shown to be true: $g^{*}(Mod(\Delta \cup \psi_{\emptyset})) \subseteq Mod(\Delta \cup \psi_{\emptyset})$ by contraction, and $Mod(\Delta \cup \psi_{\emptyset}) \subseteq Mod(\Delta)$.

The second fact is more complicated. For simplicity, we will only prove that $g^{*}(Mod(\Delta \cup \psi_{\emptyset})) \not\subseteq Mod(\alpha)$, as the proof for $g^{*}(Mod(\Delta \cup \psi_{\emptyset})) \not\subseteq Mod(\neg\alpha)$ is symmetric.

Say instead that $g^*(Mod(\Delta \cup \psi_\emptyset)) \subseteq Mod(\alpha)$. This means that

$$(\forall (m_1, m_2) \in g^*(Mod(\Delta \cup \psi_\emptyset)))[(m_1, m_2) \models_{1,Ab} \alpha], \tag{A.33}$$

or that

$$(\forall (m_1, m_2) \in g^*(Mod(\Delta \cup \psi_\emptyset)))[(m_1, m_2) \models_{1,Ab} Ab(\ell^-)] \tag{A.34}$$

From this we can infer that

$$(\forall (n_1, n_2) \models_{1,Ab} \Delta \cup \psi_\emptyset)[(n_1, n_2) \models_{1,Ab} Ab(\ell^-)], \tag{A.35}$$

by the following argument: pick any $(n_1, n_2) \models_{1,Ab} \Delta \cup \psi_\emptyset$. By Corollary 8.3.1, there is a $(o_1, o_2) \in g^*(Mod(\Delta \cup \psi_\emptyset))$ such that $o_2 \leq_{Ab} n_2$. Since $(o_1, o_2) \in g^*(Mod(\Delta \cup \psi_\emptyset))$, $(o_1, o_2) \models_{1,Ab} \alpha$, and therefore $(o_1, o_2) \models_{1,Ab} Ab(\ell^-)$. Since $o_2 \leq_{Ab} n_2$, this means $n_2 \models_{Ab} Ab(\ell^-)$.

Now recall we assumed that $\alpha, \neg\alpha \notin Th(f^*(Mod(\Delta)))$. This means that

$$\begin{array}{l} (\exists (m_1^-, m_2^-) \in f^*(Mod(\Delta)))[(m_1,^-, m_2^-) \models_{1,Ab} \neg\alpha] \\ (\exists (m_1^+, m_2^+) \in f^*(Mod(\Delta)))[(m_1,^+, m_2^+) \models_{1,Ab} \alpha] \end{array} \tag{A.36}$$

Since f^* obeys contraction, $(m_1^-, m_2^-), (m_1^+, m_2^+) \in Mod(\Delta)$.

Now from m_2^- construct another model m_2^* such that everything is the same, except that $Ab^{m_2^*} = (Ab^{m_2^-} \setminus \{\ell^-\}) \cup \{\ell^+\}$. By the Upwardly Free *Ab* Lemma, we know that $(m_1^-, m_2^*) \models_{1,Ab} \Delta$, since ℓ^+, ℓ^- are not mentioned in Δ. Furthermore, $(m_1^-, m_2^*) \models_{1,Ab} \psi_\emptyset(\alpha)$, as $(m_1^-, m_2^*) \models_{1,Ab} \neg\alpha \wedge Ab(\ell^+)$. But recall that by definition $(m_1^-, m_2^*) \models_{1,Ab} \neg Ab(\ell^-)$.

But now we have our result that $g^*(Mod(\Delta \cup \psi_\emptyset)) \not\subseteq Mod(\alpha)$, as (m_1^-, m_2^*) directly contradicts (A.35).

□

A.15 Proof of the Full Addition Theorem

Full Addition of $S_{1,Ab}$ Theorem. *Let Γ_1 be any $\mathcal{L}_1$-theory with $\Upsilon \subseteq \Gamma_1$ a consistent subset such that $\Gamma_1 \setminus \Upsilon$ is finite. Set $\Psi_{\mathcal{E}}$ to be $Elab^+(\mathcal{L}_1, \mathcal{L}_{Ab})$. Let f be the choice function underlying $\mid\!\sim_1$.*

Say f satisfies contraction, coherence, faithfulness, and α-reflection for some $\mathcal{L}_1$-formula α. Then we have $fully\text{-}addable(\alpha, \langle \mathcal{L}_{1,Ab},\ \vdash_{1,Ab},\ \mid\!\sim_{1,Ab},\ \Gamma_{1,Ab}(\Upsilon)\rangle)$.

Proof. Referring back to (5.17), We must show that

$$\begin{aligned}
&(\forall \Psi \in \Psi_{\mathcal{E}})[Consis(\alpha, \Psi \cup \Gamma_{1,Ab}(\Upsilon)) \implies \\
&\quad (\exists \psi_+ \in \Psi_{\mathcal{E}})[Consis(\psi_+, \Psi \cup \Gamma_{1,Ab}(\Upsilon)) \wedge \\
&\qquad\qquad \psi_+ \cup \Psi \cup \Gamma_{1,Ab}(\Upsilon) \mid\!\sim_{1,Ab} \alpha \wedge \\
&\qquad\qquad fully\text{-}retractable(\alpha, \langle \mathcal{L}_{1,Ab},\ \vdash_{1,Ab},\ \mid\!\sim_{1,Ab},\ \Gamma_{1,Ab}(\Upsilon)\rangle) \wedge \\
&\qquad\qquad Consis(\neg\alpha, \Psi \cup \Gamma_{1,Ab}(\Upsilon)) \implies Consis(\neg\alpha, \psi_+ \cup \Psi \cup \Gamma_{1,Ab}(\Upsilon))]]
\end{aligned} \tag{A.37}$$

Take Ψ to be an arbitrary member of $Elab^+(\mathcal{L}_1, \mathcal{L}_{Ab})$. Rewriting $\Psi \cup \Gamma_{1,Ab}(\Upsilon)$ as Δ, and using the definition of $Consis$, we end up having to show that:

$$\begin{aligned}
&(\exists (m_1, m_2) \in f^*(Mod(\Delta)))[(m_1, m_2) \models_{1,Ab} \alpha] \implies \\
&\quad (\exists \psi_+ \in \Psi_{\mathcal{E}})[\ \ (\exists (n_1, n_2) \in f^*(Mod(\Delta)))[(n_1, n_2) \models_{1,Ab} \psi_+] \wedge \\
&\qquad\qquad (\forall (o_1, o_2) \in f^*(g^*(Mod(\psi_+ \cup \Delta))))[(o_1, o_2) \models_{1,Ab} \alpha] \wedge \\
&\qquad\qquad fully\text{-}retractable(\alpha, \langle \mathcal{L}_{1,Ab},\ \vdash_{1,Ab},\ \mid\!\sim_{1,Ab},\ \psi_+ \cup \Delta\rangle) \wedge \\
&\qquad\qquad (\exists (p_1, p_2) \in f^*(Mod(\Delta)))[(p_1, p_2) \models_{1,Ab} \neg\alpha] \implies \\
&\qquad\qquad\quad (\exists (r_1, r_2) \in f^*(Mod(\psi_+ \cup \Delta)))[(r_1, r_2) \models_{1,Ab} \neg\alpha]]
\end{aligned} \tag{A.38}$$

Call $(m_1, m_2) \in f^*(Mod(\Delta))$ such that $(m_1, m_2) \models_{1,Ab} \alpha$. Let ℓ^α be a fresh symbol not mentioned in $\Psi \wedge AB(\Gamma_1 \setminus \Upsilon)$, which is $>$-than any other element of $Param$ mentioned. This is possible as $\Psi \cup AB(\Gamma_1 \setminus \Upsilon)$ is finite. Define ψ^+ as:

$$\psi^+ \equiv_{def} (Ab(\ell^\alpha) \vee \alpha) \wedge (\forall_{Ab}\ x)[Ab(\ell^\alpha) \wedge x < \ell^\alpha \implies Ab(x)] \tag{A.39}$$

Note $\psi^+ \in Elab^+(\mathcal{L}_1, \mathcal{L}_{Ab})$. To prove full addition, we have to show four facts:

1. $(\exists (n_1, n_2) \in f^*(Mod(\Delta)))[(n_1, n_2) \models_{1,Ab} \psi_+]$: From (m_1, m_2) construct m_2^* which is the same as m_2, except that $Ab^{m_2^*} = Ab^{m_2} \setminus \{\ell^\alpha\}$. By the Upwardly Free Ab Lemma, since $\Delta = \Psi \cup \Upsilon \cup AB(\Gamma_1 \setminus \Upsilon) \cup \Gamma_{Ab}$, and since ℓ^α is not mentioned in $\Psi \cup AB(\Gamma_1 \setminus \Upsilon)$, and is $>$ than those labels, we know that $(m_1, m_2^*) \in Mod(\Delta)$. By Right Interchangeability, $(m_1, m_2^*) \in f^*(Mod(\Delta))$. Finally, by construction $(m_1, m_2^*) \models_{1,Ab} \psi_+$, as $(m_1, m_2^*) \models_{1,Ab} \alpha \wedge \neg Ab(\ell^\alpha)$.

2. $(\forall (o_1, o_2) \in f^*(g^*(Mod(\psi_+ \cup \Delta))))[(o_1, o_2) \models_{1,Ab} \alpha]$: First we must show that $f^*(g^*(Mod(\psi_+ \cup \Delta)))$ is non-empty. By faithfulness of f^*, it is enough to show that $g^*(Mod(\psi_+ \cup \Delta))$ is non-empty. $\psi_+ \cup \Delta = \psi_+ \cup \Psi \cup \Upsilon \cup AB(\Gamma_1 \setminus \Upsilon) \cup \Gamma_{Ab}$. By Corollary 8.3.2, it is enough to show that Υ combined with the $\mathcal{L}_1$ conjunct $\phi \in \Psi$ is consistent. But this holds because our given witness $(m_1, m_2) \models_{1,Ab} \Delta$.

   So now, say $(o_1, o_2) \in f^*(g^*(Mod(\psi_+ \cup \Delta)))$ but $(o_1, o_2) \models_{1,Ab} \neg\alpha$. This means $(o_1, o_2) \models_{1,Ab} Ab(\ell^\alpha) \wedge (\forall_{Ab}\ x)[x < \ell^\alpha \implies Ab(x)]$. Also, $(o_1, o_2) \in g^*(Mod(\psi_+ \cup \Delta))$ by contraction of f^*, so that (o_1, o_2) obeys:

$$
\begin{aligned}
& (\forall (a_1, a_2) \in Mod(\psi_+ \cup \Delta))[\neg(a_2 <_{Ab} o_2)] \\
\equiv\ & (\forall (a_1, a_2))[((a_1, a_2) \models_{1,Ab} \psi_+ \cup \Delta \wedge \\
& \quad |a_2| = |o_2| \wedge <^{a_2} = <^{o_2} \wedge \\
& \quad (\forall \ell \in Param)[\ell^{a_2} = \ell^{o_2}] \wedge \\
& \quad Ab^{a_2} \subseteq Ab^{o_2}) \implies \\
& Ab^{a_2} = Ab^{o_2}]
\end{aligned}
\tag{A.40}
$$

   With this information about (o_1, o_2) we can construct a counter-model to (A.40). Define o_2^* to be the same as o_2, except $Ab^{o_2^*} = Ab^{o_2} \setminus \{\ell^\alpha\}$. Note that if ℓ^{max} is the highest symbol mentioned in Ψ, o_2^* satisfies $(\forall_{Ab}\ x)[x \leq \ell^{max} \implies Ab(x)]$. Now consider the model (m_1, o_2^*). $(m_1, o_2^*) \models_{1,Ab} \alpha \wedge \neg Ab(\ell^\alpha)$, so $(m_1, o_2^*) \models_{1,Ab} \psi_+$. By the Downwardly Free Ab Lemma, $(m_1, o_2^*) \models_{1,Ab} \Delta$.

Now, from the construction of o_2^*, we see that (m_1, o_2^*) satisfies the antecedent of (A.40), which means that $Ab^{o_2^*} = Ab^{o_2}$, a contradiction.

Therefore $(o_1, o_2) \models_{1,Ab} \alpha$.

3. $fully\text{-}retractable(\alpha, \langle \mathcal{L}_{1,Ab},\ \vdash_{1,Ab},\ \mid\!\sim_{1,Ab},\ \psi_+ \cup \Delta\rangle)$:

   By the Full Retractability Lemma, all we have to show is that $\psi_+ \cup \Delta$ is of the proper form. $\psi_+ \in Elab^+(\mathcal{L}_1, \mathcal{L}_{Ab})$, $\Delta = \Psi \cup \Upsilon \cup AB(\Gamma_1 \setminus \Upsilon) \cup \Gamma_{Ab}$ is also of the correct form as Υ is an $\mathcal{L}_1$-theory, and $\Psi \wedge AB(\Gamma_1 \setminus \Upsilon) \in Elab^+(\mathcal{L}_1, \mathcal{L}_{Ab})$, being finite.

4. $(\exists (p_1, p_2) \in f^*(Mod(\Delta)))(p_1, p_2) \models_{1,Ab} \neg\alpha] \implies$

   $(\exists (r_1, r_2) \in f^*(Mod(\psi_+ \cup \Delta)))[(r_1, r_2) \models_{1,Ab} \neg\alpha]$. Since $Mod(\psi_+ \cup \Delta) \subseteq Mod(\Delta)$, coherence lets us know that

   $$Mod(\psi_+ \cup \Delta) \cap f^*(Mod(\Delta)) \subseteq f^*(Mod(\psi_+ \cup \Delta)) \tag{A.41}$$

   Hence, it is enough to show that for some p_2^*, $(p_1, p_2^*) \in Mod(\psi_+ \cup \Delta) \cap f^*(Mod(\Delta))$. Let p_2^* be like p_2, except $Ab^{p_2^*} = |p_2^*|$. Since $(p_1, p_2) \in Mod(\Delta)$, by the Downwardly Free *Ab* Lemma $(p_1, p_2^*) \in Mod(\Delta)$, but also $(p_1, p_2^*) \in Mod(\psi_+)$.

   To show $(p_1, p_2^*) \in f^*(Mod(\Delta))$, it is enough to show that $p_1 \in f([Mod(\Delta)]_\ell)$, which holds by assumption.

□

A.16 Proof of Theorem 9.2.1

Theorem 9.2.1 [Satisfiability is Preserved]. *Let* $S_1 = \langle \mathcal{L}_1,\ \vdash_1,\ \mid\!\sim_1,\ \Gamma_1\rangle$ *be an extended axiomatic formal system with choice. Say that* $\Upsilon \subseteq \Gamma_1$ *is satisfiable. Then the combined system* $S_{1,Ab}$ *has an* $\mathcal{L}_{1,Ab}$*-model.*

Proof. We must show that there is a model of $\Gamma_{1,Ab}(\Upsilon) = \Upsilon \cup AB(\Gamma_1 \setminus \Upsilon) \cup \Gamma_{Ab}$. Let m_1 be the $\mathcal{L}_1$-model of Υ. Let m_2 be a model of Γ_{Ab}, with $Ab^{m_2} = |m_2|$. We argue that $(m_1, m_2) \models_{1,Ab} \Gamma_{1,Ab}(\Upsilon)$:

$(m_1, m_2) \models_{1,Ab} \Upsilon$ by construction of m_1, and $(m_1, m_2) \models_{1,Ab} \Gamma_{Ab}$ because of m_2's construction. Finally, $(m_1, m_2) \models AB(\Gamma_1 \setminus \Upsilon)$ trivially as Ab is universally true. □

A.17 Proof of Theorem 9.2.2

Theorem 9.2.2 [Hard Truths are Preserved by Hard Consequence]. *Let α be an $\mathcal{L}_1$-formula and say $\Upsilon \subseteq \Gamma_1$ and $\Upsilon \vdash_1 \alpha$. Then $\Gamma_{1,Ab}(\Upsilon) \vdash_{1,Ab} \alpha$.*

Proof. We need to show that $\Upsilon \cup AB(\Gamma_1 \setminus \Upsilon) \cup \Gamma_{Ab} \vdash_{1,Ab} \alpha$. Let $(m_1, m_2) \in f^*(Mod(\Upsilon \cup AB(\Gamma_1 \setminus \Upsilon) \cup \Gamma_{Ab}))$. We must show $(m_1, m_2) \models_{1,Ab} \alpha$. We are given that $(\forall (a_1, a_2) \models_{1,Ab} \Upsilon)[(a_1, a_2) \models_{1,Ab} \alpha]$. By contraction, $f^*(Mod(\Upsilon \cup AB(\Gamma_1 \setminus \Upsilon) \cup \Gamma_{Ab})) \subseteq Mod(\Upsilon \cup AB(\Gamma_1 \setminus \Upsilon) \cup \Gamma_{Ab}) \subseteq Mod(\Upsilon)$, so this follows. □

A.18 Proof of Lemma 9.2.1

Lemma 9.2.1 [g^*'s Behavior when Γ_1 is satisfiable.]. *Say Γ_1 is satisfiable. Then*

$$(m_1, m_2) \in g^*(Mod(\Gamma_{1,Ab}(\Upsilon))) \iff \begin{array}{l} m_1 \models_1 \Gamma_1 \wedge \\ Ab^{m_2} = \emptyset \wedge \\ m_2 \models_2 \Gamma_{Ab} \end{array} \tag{A.42}$$

Proof. $\rightarrow$: Assume $(m_1, m_2) \in g^*(Mod(\Gamma_{1,Ab}(\Upsilon)))$. It is enough to show that $Ab^{m_2} = \emptyset$ and $m_2 \models_{1,Ab} \Gamma_{Ab}$, as since we are given that $(m_1, m_2) \models_{1,Ab} \Upsilon \cup AB(\Gamma_1 \setminus \Upsilon)$, $Ab^{m_2} = \emptyset$ means $(m_1, m_2) \models_{1,Ab} \Gamma_1$.

By contraction of g^*, $(m_1, m_2) \models_{1,Ab} \Upsilon \cup AB(\Gamma_1 \setminus \Upsilon) \cup \Gamma_{Ab}$. Hence we automatically get that $m_2 \models_{1,Ab} \Gamma_{Ab}$. From the definition of g^* we know that

$$(\forall (a_1, a_2) \in Mod(\Gamma_{1,Ab}(\Upsilon)))[\neg a_2 <_{Ab} m_2] \tag{A.43}$$

Now say instead there is some ℓ such that that $(m_1, m_2) \models_{1,Ab} Ab(\ell)$. Consider an $\mathcal{L}_1$-structure n_1 such that $n_1 \models_1 \Gamma_1$. Let n_2 be the same as m_2, except that $Ab^{n_2} = \emptyset$. $(n_1, n_2) \models_{1,Ab} \Gamma_1 \wedge \Gamma_{Ab}$, so $(n_1, n_2) \models_{1,Ab} \Gamma_{1,Ab}(\Upsilon)$.

But clearly, (n_1, n_2) applied to (A.43) results in a contradiction, as $n_2 <_{Ab} m_2$. Hence we must conclude that in fact $Ab^{m_2} = \emptyset$.

$\leftarrow$: Say $(m_1, m_2) \models_{1,Ab} \Gamma_1$ and $Ab^{m_2} = \emptyset$ and $m_2 \models_2 \Gamma_{Ab}$. We must show that $(m_1, m_2) \models_{1,Ab} \Gamma_{1,Ab}(\Upsilon)$, which is clear, and that (A.43) holds. So let $(a_1, a_2) \in Mod(\Gamma_{1,Ab}(\Upsilon))$. But say in fact that $a_2 <_{Ab} m_2$. This entails that $Ab^{a_2} \subset Ab^{m_2} = \emptyset$, which is impossible. $\square$

A.19 Proof of Theorem 9.2.3

Theorem 9.2.3 [Preservation of Soft Formulas]. *Let α be an $\mathcal{L}_1$-formula. Say Γ_1 is satisfiable and that $\Gamma_1 \mid\!\sim_1 \alpha$. Then $\Gamma_{1,Ab}(\Upsilon) \mid\!\sim_{1,Ab} \alpha$.*

Proof. We are given that:

$$(\forall m_1 \in f(Mod(\Gamma_1)))[m_1 \models_1 \alpha] \tag{A.44}$$

Let $(n_1, n_2) \in f^*(g^*(Mod(\Gamma_{1,Ab}(\Upsilon))))$. We must show that $(n_1, n_2) \models_{1,Ab} \alpha$. It is enough to show that $[f^*(g^*(Mod(\Gamma_{1,Ab}(\Upsilon))))]_\ell \subseteq f(Mod(\Gamma_1))$. This means we have to show that $(a, b) \in f^*(g^*(Mod(\Gamma_{1,Ab}(\Upsilon)))) \implies a \in f(Mod(\Gamma_1))$, or unraveling the definition of f^*:

$$\begin{aligned}&[(a, b) \in g^*(Mod(\Gamma_{1,Ab}(\Upsilon)))\wedge \\ &a \in f(\{s \mid (s, t) \in g^*(Mod(\Gamma_{1,Ab}(\Upsilon)))\})] \quad \implies \quad a \in f(Mod(\Gamma_1))\end{aligned} \tag{A.45}$$

With (A.45), it is enough to show that $\{s \mid (s, t) \in g^*(Mod(\Gamma_{1,Ab}(\Upsilon)))\} = Mod(\Gamma_1)$.

$\subseteq$: Say $(s, t) \in g^*(Mod(\Gamma_{1,Ab}(\Upsilon)))$. Then by Lemma 9.2.1, we get $s \models_1 \Gamma_1$.

$\supseteq$: Say $s \models_1 \Gamma_1$. Let t be an $\mathcal{L}_{Ab}$-structure which is a model of Γ_{Ab} such that $Ab^t = \emptyset$. Then again by Lemma 9.2.1 we see that $(s, t) \in g^*(Mod(\Gamma_{1,Ab}(\Upsilon)))$. $\square$

A.20 Proof of Corollary 9.3.1

Corollary 9.3.1 [Classical FOL Has Full Retraction and Addition]. *Let* Γ_1 *be a theory in some first-order language* $\mathcal{L}_1$*. Say* $\Upsilon \subseteq \Gamma_1$ *is some consistent subset such that* $\Gamma_1 \setminus \Upsilon$ *is finite. Then the system* $\langle \mathcal{L}_{1,Ab},\ \vdash_{1,Ab},\ \mid\!\sim_{1,Ab},\ \Gamma_{1,Ab}(\Upsilon)\rangle$ *where* $f = \mathbf{1}$ *has full retraction and full addition, for any* $\alpha \in \mathcal{L}_1$*.*

Proof. All we have to show is that $f = \mathbf{1}$ satisfies coherence, contraction, faithfulness and α-reflection for arbitrary $\alpha \in \mathcal{L}_1$.

1. Coherence: $\mathbf{1}(X) = X \subseteq X$.
2. Contraction: Say $X \subseteq Y$. $X \cap \mathbf{1}(Y) = X \cap Y \subseteq \mathbf{1}(X) = X$.
3. Faithfulness: Say $\mathbf{1}(X) = \emptyset$. Then $X = \emptyset$.
4. α-reflection: Say $Y \subseteq X$, $\alpha, \neg\alpha \notin Th(\mathbf{1}(X))$, and $\alpha, \neg\alpha \notin Th(Y)$. Then trivially $\alpha, \neg\alpha \notin Th(\mathbf{1}(Y))$.

□

A.21 Proof of Corollary 9.3.2

Corollary 9.3.2 [Preferential Semantics and Full Retraction and Addition]. *Let* $\langle \mathcal{L}_1,\ \vdash_1,\ \mid\!\sim_1,\ \Gamma_1(\Upsilon)\rangle$ *be an extended axiomatic formal system with choice function* f*. Say* $f(X) =_{def} \{w \in X \mid (\forall x \in X)[x \leq w \implies x = w]\}$ *for some well-founded relation* $\leq$*. Furthermore assume that* f *satisfies* α*-reflection.*

Say $\Upsilon \subseteq \Gamma_1$ *is some consistent subset such that* $\Gamma_1 \setminus \Upsilon$ *is finite. Then the system* $\langle \mathcal{L}_{1,Ab},\ \vdash_{1,Ab},\ \mid\!\sim_{1,Ab},\ \Gamma_{1,Ab}(\Upsilon)\rangle$ *has full retraction and full addition for* α*.*

Proof. All we have to show is that f satisfies coherence, contraction, and faithfulness. The first two properties hold by Proposition 5.3.1 for item 3.

As for faithfulness, say $f(X) = \emptyset$. This is either because $X = \emptyset$, or $(\forall w \in X)(\exists x \in X)[x \leq w \implies x \neq w]$. But this contradicts the well-foundedness of $\leq$. □

A.22 Proof of the $[f^*]_\ell$-Equivalence Lemma

$[f^*]_\ell$-Equivalence Lemma. *Say that $[X]_\ell = [Y]_\ell$. Then $[f^*(X)]_\ell = [f^*(Y)]_\ell$.*

Proof. Without loss of generality it is enough to show that $[f^*(X)]_\ell \subseteq [f^*(Y)]_\ell$. Say $a \in [f^*(X)]_\ell$, This means that $(a,b) \in f^*(X)$, or that $(a,b) \in X$ and $a \in f([X]_\ell)$.

To show $a \in [f^*(Y)]_\ell$, we must show that $(a,d) \in f^*(Y)$ for some d. This means that we must have that $(a,d) \in Y$ and $a \in f^*([Y]_\ell)$ for some d. But these both hold because $[X]_\ell = [Y]_\ell$: the first because $(a,b) \in X$ means $a \in [X]_\ell$ so that $a \in [Y]_\ell$ and so for some d, $(a,d) \in Y$. The second fact follows by substitution. □

A.23 Proof of Theorem 9.7.1

Lemma A.23.1 (Restricting models to *Param*). *Let Ψ be a set of sentences of the form:*

$$\begin{array}{ll} \phi, & \text{where } \phi \in \mathcal{L}_1 \\ Ab(\ell) \vee \phi, & \text{where } \phi \in \mathcal{L}_1, \text{and } \ell \in Param \\ (\forall_{Ab}\ x)[Ab(\ell) \wedge x < \ell \implies Ab(x)] & \text{where } \ell \in Param \end{array} \tag{A.46}$$

If $(m_1, m_2) \models_{1,Ab} \Psi \cup \Gamma_{Ab}$ then $m_1, m_2' \models_{1,Ab} \Psi \cup \Gamma_{Ab}$, where m_2' is just like m_2, except its interpretation is restricted to symbols of Param. In other words, $|m_2'| = \{\ell^{m_2} \mid \ell \in Param\}$.

Proof. Go ahead and assume $(m_1, m_2) \models_{1,Ab} \Psi \cup \Gamma_{Ab}$. It is clear that $(m_1, m_2') \models_{1,Ab} \Gamma_{Ab}$.

For each $\phi \in \Psi$, we know that $m_1, m_2' \models_{1,Ab} \phi$ as it only depends on m_1.

For each $Ab(\ell) \vee \phi \in \Psi$, if $(m_1, m_2) \models_{1,Ab} Ab(\ell) \vee \phi$, then so does (m_1, m_2') – say if $m_1 \models_1 \neg\phi$, then $m_2 \models_{Ab} Ab(\ell)$, and then $m_2' \models_{Ab} Ab(\ell)$ as well.

Finally, for each $(\forall_{Ab}\ x)[Ab(\ell) \wedge x < \ell \implies Ab(x)] \in \Psi$, assume $(m_1, m_2) \models_{1,Ab} (\forall_{Ab}\ x)[Ab(\ell) \wedge x < \ell \implies Ab(x)] \in \Psi$, and show $(m_1, m_2') \models_{1,Ab} (\forall_{Ab}\ x)[Ab(\ell) \wedge x < \ell \implies Ab(x)] \in \Psi$. Say $(m_1, m_2') \models_{1,Ab} Ab(\ell) \wedge a < \ell$ for some $a \in |m_2'|$. Hence

$m_2 \models_{Ab} Ab(a)$. Now, $a = \ell_0^{m_2}$ for some $\ell_0 \in Param$, so $a \in |m_2'|$, and therefore $m_2' \models_{Ab} Ab(a)$. □

Lemma A.23.2 **($Mod(\Sigma) \subseteq Mod(Compaction_1(\Sigma))$).** *Let Σ be as described in (9.13).*

$$Mod(\Sigma) \subseteq Mod(Compaction_1(\Sigma)) \tag{A.47}$$

Proof. Say $(m_1, m_2) \models_{1,Ab} \Sigma$. We only need to show that $(m_1, m_2) \models_{1,Ab} Ab(\ell_k) \implies Ab(\ell)$ for any $k \in K \wedge \ell \in \Sigma$ such that $\ell < \ell_k \in \Gamma_{Ab}$.

So go ahead and assume that $(m_1, m_2) \models_{1,Ab} Ab(\ell_k)$. This means that $(m_1, m_2) \models_{1,Ab} (\forall_{Ab}\ x)[x < \ell_k \implies Ab(x)]$. Since $\ell < \ell_k \in \Gamma_{Ab}$, $(m_1, m_2) \models_{1,Ab} \ell < \ell_k$, so $(m_1, m_2) \models_{1,Ab} Ab(\ell)$. □

Lemma A.23.3 **($[g^*(Mod(\Psi' \cup \Sigma))]_\ell = [g^*(Mod(\Psi' \cup Compaction_1(\Sigma)))]_\ell$).** *Let Σ be of the form above, and $\Psi' \in Elab^+(\mathcal{L}_1, \mathcal{L}_{Ab})$. Then*

$$[g^*(Mod(\Psi' \cup \Sigma))]_\ell = [g^*(Mod(\Psi' \cup Compaction_1(\Sigma)))]_\ell, \tag{A.48}$$

Proof. $\subseteq$: Say $(m_1, m_2) \in g^*(Mod(\Psi' \cup \Sigma))$. We must show that $(m_1, m_2^*) \in g^*(Mod(\Psi' \cup Compaction_1(\Sigma)))$ for some m_2^*. Define the $\mathcal{L}_{Ab}$-structure m_2^* to be the same as m_2, except $|m_2^*| = \{\ell^{m_2} \mid \ell \in Param\}$.

We are given that

$$\begin{aligned} &(m_1, m_2) \models_{1,Ab} \Psi' \cup \Sigma, \text{ and} \\ &(\forall (n_1, n_2) \models_{1,Ab} \Psi' \cup \Sigma)[\neg n_2 <_{Ab} m_2] \end{aligned} \tag{A.49}$$

We must show that

1. $(m_1, m_2^*) \models_{1,Ab} \Psi' \cup Compaction_1(\Sigma)$ and
2. $(\forall (o_1, o_2) \models_{1,Ab} \Psi' \cup Compaction_1(\Sigma))[\neg o_2 <_{Ab} m_2^*]$.

The first task is straightforward. $(m_1, m_2^*) \models_{1,Ab} \Psi' \cup \Sigma$ by Lemma A.23.1. Then by Lemma A.23.2 we know that $(m_1, m_2^*) \models_{1,Ab} \Psi' \cup Compaction_1(\Sigma)$.

As for the second item, assume instead there is some $(o_1, o_2) \models_{1,Ab} \Psi' \cup \mathit{Compaction}_1(\Sigma)$ but in fact, $o_2 <_{Ab} m_2^*$. Call ℓ^* the $>$-largest element in $Ab^{m_2^*} \setminus Ab^{o_2}$ which must be an element of Param. Construct o_2^* to be the same as m_2, except $Ab^{o_2^*} = Ab^{m_2} \setminus \ell^*$. Clearly $o_2^* <_{Ab} m_2$. If we can show $(o_1, o_2^*) \models_{1,Ab} \Psi' \cup \Sigma$ then we are done, because this contradicts the fact that (m_1, m_2) is Ab-minimal as per (A.49).

There are three kinds of formulas in $\Psi' \cup \Sigma$:

1. If $\phi \in \Psi' \cup \Sigma$ then $\phi \in \Psi' \cup \mathit{Compaction}_1(\Sigma)$, so that $(o_1, o_2^*) \models_{1,Ab} \phi$ as $o_1 \models_1 \phi$ from our assumption.

2. If $Ab(\ell_i) \vee \phi \in \Psi' \cup \Sigma$, then $Ab(\ell_i) \vee \phi \in \Psi' \cup \mathit{Compaction}_1(\Sigma)$, and if $(o_1, o_2^*) \models_{1,Ab} \neg\phi$, $(o_1, o_2) \models_{1,Ab} \neg\phi$, so $(o_1, o_2) \models_{1,Ab} Ab(\ell_i)$. $Ab^{o_2} \subset Ab^{m_2^*} \subseteq Ab^{m_2}$, so $m_2 \models_{Ab} Ab(\ell_i)$. The only way $o_2^* \not\models_{Ab} Ab(\ell_i)$ is if $\ell_i = \ell^*$. But this would mean that $o_2 \models_{Ab} \neg Ab(\ell_i)$. But $(o_1, o_2) \models_{1,Ab} Ab(\ell_i) \vee \phi$ which means $o_1 \models_1 \phi$, a contradiction.

3. If $(\forall_{Ab}\, x)[Ab(\ell_j) \wedge x < \ell_j \implies Ab(x)] \in \Psi' \cup \Sigma$. Say $(o_1, o_2^*) \models_{1,Ab} Ab(\ell_j) \wedge a < \ell_j$ for some $a \in |o_2^*|$. We need to show that $(o_1, o_2^*) \models_{1,Ab} Ab(a)$. Since o_2^* looks just like m_2, and $o_2^* <_{Ab} m_2$, we know that $m_2 \models_{Ab} Ab(\ell_j) \wedge a < \ell_j$, and hence $m_2 \models_{Ab} Ab(a)$.

   To show $o_2^* \models Ab(a)$ it is enough to show that $a \neq \ell^*$. Say in fact it was. Since $m_2 \models_{Ab} Ab(\ell_j) \wedge \ell^* < \ell_j$, so will $m_2^* \models_{Ab} Ab(\ell_j) \wedge \ell^* < \ell_j$. $\Gamma_{Ab} \subseteq \mathit{Compaction}_1(\Sigma)$ is complete, so we know that $\ell^* < \ell_j \in \Gamma_{Ab}$, and since $(o_1, o_2) \models \Psi' \cup \mathit{Compaction}_1(\Sigma)$, so $(o_1, o_2) \models Ab(\ell_j) \implies Ab(\ell^*)$. Since $o_2 \models_{Ab} \neg Ab(\ell^*)$, $o_2 \models_{Ab} \neg Ab(\ell_j)$. But note that we have shown that $m_2 \models_{Ab} Ab(\ell_j)$, so that $\ell_j \in Ab^{m_2^*} \setminus Ab^{o_2}$, but that contradicts the fact that ℓ^* is the $>$-largest element in $Ab^{m_2^*} \setminus Ab^{o_2}$.

$\supseteq$: Say $(m_1, m_2) \in g^*(\Psi' \cup \mathit{Compaction}_1(\Sigma))$. We need to show that $(m_1, m_2^*) \in g^*(\Psi' \cup \Sigma)$ for some m_2^*. We are given that

$$
\begin{aligned}
&(m_1, m_2) \models_{1,Ab} \Psi' \cup \mathit{Compaction}_1(\Sigma) \\
&(\forall (n_1, n_2) \models_{1,Ab} \Psi' \cup \mathit{Compaction}_1(\Sigma))[\neg n_2 < m_2]
\end{aligned}
\tag{A.50}
$$

We must show that

1. $(m_1, m_2^*) \models_{Ab} \Psi' \cup \Sigma$ and
2. $(\forall (o_1, o_2) \models_{1,Ab} \Psi' \cup \Sigma)[\neg o_2 <_{Ab} m_2^*]$

Define m_2^* to look like m_2, except restricted to the domain of *Param*: $|m_2|^* = \{\ell^{m_2} \mid \ell \in Param\}$. Let us show that $(m_1, m_2^*) \models_{Ab} \Psi' \cup \Sigma$ first. There are three kinds of formulas in $\Psi' \cup \Sigma$:

1. Say $\phi \in \Psi' \cup \Sigma$. Then $\phi \in \Psi' \cup Compaction_1(\Sigma)$, and since $(m_1, m_2) \models_{1,Ab} \Psi' \cup Compaction_1(\Sigma)$, $m_1 \models_1 \phi$, so $(m_1, m_2^*) \models_1 \phi$.
2. Say $Ab(\ell_i) \vee \phi_i \in \Psi' \cup \Sigma$. $Ab(\ell_i) \vee \phi_i \in \Psi' \cup Compaction_1(\Sigma)$, and so $(m_1, m_2) \models_{1,Ab} Ab(\ell_i) \vee \phi_i$. We need to show $(m_1, m_2^*) \models_{1,Ab} Ab(\ell_i) \vee \phi_i$, so assume $(m_1, m_2^*) \models_{1,Ab} \neg\phi_i$. Hence $(m_1, m_2) \models_{1,Ab} \neg\phi_i$, and thus $(m_1, m_2) \models_{1,Ab} Ab(\ell_i)$. $a \in Ab^{m_2^*} \iff a \in Ab^{m_2} \wedge a = \ell^{m_2}$, so we can infer that $(m_1, m_2^*) \models_{1,Ab} Ab(\ell_i)$.
3. Say $(\forall_{Ab}\ x)[Ab(\ell_j) \wedge x < \ell_j \implies Ab(x)] \in \Psi' \cup \Sigma$. Say $(m_1, m_2^*) \models_{1,Ab} Ab(\ell_j) \wedge a < \ell_j$ for some $a \in |m_2^*|$. Since $a \in |m_2^*|$, there must be some $\ell_a \in Param$ such that $\ell_a^{m_2} = a$. We can also say that $m_2 \models_{Ab} Ab(\ell_j) \wedge \ell_a < \ell_j$. We need to show that $(m_1, m_2^*) \models_{1,Ab} Ab(\ell_a)$. There are two cases:
   (a) $(\forall_{Ab}\ x)[Ab(\ell_j) \wedge x < \ell_j \implies Ab(x)] \in \Psi'$: This means $(m_1, m_2) \models_{1,Ab} (\forall_{Ab}\ x)[Ab(\ell_j) \wedge x < \ell_j \implies Ab(x)]$. Hence $(m_1, m_2) \models_{1,Ab} Ab(\ell_a)$. Then by the definition of m_2^*, $(m_1, m_2^*) \models_{1,Ab} Ab(\ell_a)$.
   (b) $(\forall_{Ab}\ x)[Ab(\ell_j) \wedge x < \ell_j \implies Ab(x)] \in \Sigma$: Since $m_2 \models_{Ab} Ab(\ell_j) \wedge \ell_a < \ell_j$, and Γ_{Ab} is complete we know that $\ell_a < \ell_j \in \Gamma_{Ab}$, and thus that $Ab(\ell_j) \implies Ab(\ell_a) \in Compaction_1(\Sigma)$. Hence $m_2 \models_{Ab} Ab(\ell_a)$, and therefore $m_2^* \models_{Ab} Ab(\ell_a)$.

Now suppose that $(o_1, o_2) \models_{1,Ab} \Psi' \cup \Sigma$ but $o_2 <_{Ab} m_2^*$. By Lemma A.23.2, we know that $(o_1, o_2) \models_{1,Ab} \Psi' \cup Compaction_1(\Sigma)$. Let ℓ^* be the $>$-largest element of

$Param$ mentioned in $Ab^{m_2^*} \setminus Ab^{o_2}$. From o_2 construct o_2^* which is just like m_2, except $Ab^{o_2^*} = Ab^{m_2} \setminus \{\ell^*\}$. $o_2^* <_{Ab} m_2$. If we can show that $(o_1, o_2^*) \models \Psi' \cup Compaction_1(\Sigma)$ we have reached a contradiction and are done.

There are four kinds of formulas in $\Psi' \cup Compaction_1(\Sigma)$:

1. $\phi \in \Psi' \cup Compaction_1(\Sigma)$. We can infer that $\phi \in \Psi' \cup \Sigma$, so that $(o_1, o_2) \models_{1,Ab} \phi$, so that so does $(o_1, o_2^*) \models_{1,Ab} \phi$ since it only depends on m_1.

2. $Ab(\ell_i) \vee \phi_i \in \Psi' \cup Compaction_1(\Sigma)$ and so $Ab(\ell_i) \vee \phi_i \in \Psi' \cup \Sigma$, and therefore $(o_1, o_2) \models_{1,Ab} Ab(\ell_i) \vee \phi_i$. To show $(o_1, o_2^*) \models_{1,Ab} Ab(\ell_i) \vee \phi_i$, assume $(o_1, o_2^*) \models_{1,Ab} \neg\phi_i$, so that $o_2 \models_{Ab} Ab(\ell_i)$. Since $o_2 <_{Ab} m_2*$, and m_2 is just m_2^* restricted to $Param$, we can infer that $m_2 \models_{Ab} Ab(\ell_i)$. The only way $o_2^* \not\models_{Ab} Ab(\ell_i)$ is if $\ell_i = \ell^*$. But then this would mean that $o_2 \models_{Ab} \neg Ab(\ell_i)$, and thus that $o_1 \models_1 \phi$, a contradiction.

3. $(\forall_{Ab}\ x)[Ab(\ell_j) \wedge x < \ell_j \implies Ab(x)] \in \Psi' \cup Compaction_1(\Sigma)$ means that $(\forall_{Ab}\ x)[Ab(\ell_j) \wedge x < \ell_j \implies Ab(x)] \in \Psi'$.

   So we know that $(o_1, o_2) \models_{1,Ab} (\forall_{Ab}\ x)[Ab(\ell_j) \wedge x < \ell_j \implies Ab(x)]$. Say $(o_1, o_2^*) \models_{1,Ab} Ab(\ell_j) \wedge a < \ell_j$ for some $a \in o_2^*$. We want to show that $o_2^* \models_{Ab} Ab(a)$. Since $o_2^* <_{Ab} m_2$, we know that $m_2 \models_{Ab} Ab(\ell_j) \wedge a < \ell_j$. $(m_1, m_2) \models_{1,Ab} \Psi'$, so $m_2 \models_{Ab} Ab(a)$. Now we just have to show that $a \neq \ell^* m_2$ to show that $o_2^* \models_{Ab} Ab(a)$. Say it was. Then since $\ell_j \neq \ell^*$, $o_2 \models_{Ab} Ab(\ell_j) \wedge \ell^* < \ell_j$, and therefore $o_2 \models_{Ab} Ab(\ell^*)$. But this contradicts the definition of ℓ^*.

4. $Ab(\ell_k) \implies Ab(\ell) \in \Psi' \cup Compaction_1(\Sigma)$. This means there was some $(\forall_{Ab}\ x)[Ab(\ell_k) \wedge x < \ell_k \implies Ab(x)] \in \Sigma$ and $\ell < \ell_k \in \Gamma_{Ab}$. We must show that $(o_1, o_2^*) \models_{1,Ab} Ab(\ell_k) \implies Ab(\ell)$. So assume $o_2^* \models_{Ab} Ab(\ell_k)$. Hence $m_2 \models_{Ab} Ab(\ell_k)$, and thus $m_2 \models_{Ab} Ab(\ell)$ since $(m_1, m_2) \models_{1,Ab} Compaction_1(\Sigma)$. If we can show $\ell \neq \ell^*$, we are done. So assume not. Then $o_2 \models_{Ab} \neg Ab(\ell^*)$, and since $o_2 \models_{Ab} \ell^* < \ell_k$, we must infer that $o_2 \models_{Ab} \neg Ab(\ell_k)$. Also $m_2^* \models_{Ab} \ell_k$. Hence $\ell_k \in Ab^{m_2^*} \setminus Ab^{o_2}$, and $\ell^* < \ell_k$, but this contradicts the fact that ℓ^* is the $>$-largest element in $Ab^{m_2^*} \setminus Ab^{o_2}$.

□

Theorem 9.7.1 [$Compaction_1$ **equivalent to** Σ **under elaborations**]. *Let* Σ *be of the form above. Then for any* $\Psi' \in Elab^+(\mathcal{L}_1, \mathcal{L}_{Ab})$ *and* $\phi \in \mathcal{L}_1$*, we have:*

$$\Psi' \cup \Sigma \mid\!\sim_{1,Ab} \phi \iff \Psi' \cup Compaction_1(\Sigma) \mid\!\sim_{1,Ab} \phi \tag{A.51}$$

Proof. It is enough to show that $[f^*(g^*(\Psi' \cup \Sigma))]_\ell = [f^*(g^*(\Psi' \cup Compaction_1(\Sigma)))]_\ell$. From the $[f^*]_\ell$-Equivalence Lemma it is enough to show that

$$[g^*(\Psi' \cup \Sigma)]_\ell = [g^*(\Psi' \cup Compaction_1(\Sigma))]_\ell, \tag{A.52}$$

But this holds thanks to Lemma A.23.3. □

Note that by example 4.4.1, $[g^*(Mod(\Delta))]_\ell \supseteq [g^*(Mod(\Delta \cup \psi_\emptyset))]_\ell$ does not generally hold.

A.24 Proof of the ψ^+ Covering Lemma

ψ^+ **Covering Lemma.** *Let* Φ *be some* $\mathcal{L}_1$*-theory,* $\Psi \in Elab^+(\mathcal{L}_1, \mathcal{L}_{Ab})$*, and* Γ_{Ab} *as defined. Say* $\Delta = \Psi \cup \Phi \cup \Gamma_{Ab}$*. Let* α *be a* $\mathcal{L}_1$*-formula and* $\psi_+ = (Ab(\ell^\alpha) \vee \alpha) \wedge (\forall_{Ab}\ x)[Ab(\ell^\alpha \wedge x < \ell^\alpha \implies Ab(x))]$*, where* ℓ^α *is* $>$*-than any labels mentioned in* Δ*.*

Then $g^*(Mod(\alpha \cup \Delta)) \subseteq g^*(Mod(\psi^+(\alpha) \cup \Delta))$*.*

Proof. Say $(m_1, m_2) \in g^*(Mod(\alpha \cup \Delta))$. This means that:

$$\begin{aligned} &(m_1, m_2) \models_{1,Ab} \alpha \cup \Delta \\ &(\forall (n_1, n_2) \models_{1,Ab} \alpha \cup \Delta)[\neg n_2 <_{Ab} m_2] \end{aligned} \tag{A.53}$$

We want to show that $(m_1, m_2) \in g^*(\psi^+(\alpha) \cup \Delta)$. for this, we have to show that

1. $(m_1, m_2) \models_{1,Ab} \psi^+(\alpha) \cup \Delta$ and that
2. $(\forall (o_1, o_2) \models_{1,Ab} \psi^+(\alpha) \cup \Delta)[\neg o_2 <_{Ab} m_2]$.

To show $(m_1, m_2) \models_{1,Ab} \psi^+(\alpha) \cup \Delta$, it is enough to show that $(m_1, m_2) \models_{1,Ab} \neg Ab(\ell^\alpha)$. But this is clear – if it did not, then we could always construct m_2^* where $Ab^{m_2^*} = Ab^{m_2} \setminus \{\ell^\alpha\}$, and $(m_1, m_2^*) \models_{1,Ab} \Delta$ by the Upwardly Free *Ab* Lemma and $m_2^* <_{Ab} m_2$, refuting (A.53).

Now, to show $(\forall (o_1, o_2) \models_{1,Ab} \psi^+(\alpha) \cup \Delta)[\neg o_2 <_{Ab} m_2]$. So say in fact that $(o_1, o_2) \models_{1,Ab} \psi^+(\alpha) \cup \Delta \wedge o_2 <_{Ab} m_2$. Since $m_2 \models_{Ab} \neg Ab(\ell^\alpha)$, then so does o_2. Hence $(o_1, o_2) \models_{1,Ab} \alpha$, and thus by (A.53), $\neg o_2 <_{Ab} m_2$, a contradiction.

$\square$

Index

Bibliography

[Alchourrón et al., 1985] Alchourrón, C. E., Gärdenfors, P., and Makinson, D. (1985). On the logic of theory change: Partial meet contraction and revision functions. *The Journal of Symbolic Logic*, 50(2):510–530.

[Alferes et al., 2000] Alferes, J. J., Leite, J. A., Pereira, L. M., Przymusinska, H., and Przymusinski, T. C. (2000). Dynamic updates of non-monotonic knowledge bases. *Journal of Logic Programming*, 45(1-3):43–70.

[Allen, 1981] Allen, J. (1981). An interval-based representation of temporal knowledge. In Hayes, P. J., editor, *IJCAI*, pages 221–226.

[Amir, 1998] Amir, E. (1998). Towards a Formalization of Elaboration Tolerance: Adding and Deleting Axioms[1]. In *Symposium on Abstraction, Reformulation and Approximation (SARA98)*. Also appeared in Seventh International Workshop on Nonmonotonic Reasoning (Belief Revision track), and extended to [Amir, 2000].

[Amir, 2000] Amir, E. (2000). *Frontiers of Belief Revision*, chapter Towards a Formalization of Elaboration Tolerance: Adding and Deleting Axioms[2]. Kluwer. Elaboration of [Amir, 1998].

[Amir, 2001] Amir, E. (2001). *Dividing and Conquering Logic*[3]. PhD thesis, Stanford University.

[1] http://www.cs.berkeley.edu/~eyal/papers/et-def-sara98.ps
[2] http://www.cs.berkeley.edu/~eyal/papers/et-def-syntactic.ps
[3] http://www.cs.berkeley.edu/~eyal/papers/phd-thesis2001.ps

[Baker, 1991] Baker, A. B. (1991). Nonmonotonic reasoning in the framework of situation calculus. *Artificial Intelligence*, 49(1-3):5–23.

[Boutilier, 1996] Boutilier, C. (1996). Iterated revision and minimal change of conditional beliefs. *Journal of Philosophical Logic*, 25(3):263–305.

[Brin and Page, 2003] Brin, S. and Page, L. (2003). Google.

[Chellas, 1980] Chellas, B. F. (1980). *Modal Logic: An Introduction*. Cambridge University Press.

[Chernoff, 1954] Chernoff, H. (1954). Rational selection of decision functions. *Econometrica*, 26:121–127.

[Chirkova, 2000] Chirkova, R. (2000). Linearly Bounded Reformulations of Conjunctive Databases[4]. In *Proceedings of the Sixth International Conference on Deductive and Object-Oriented Databases (DOOD-2000)*.

[Cohen et al., 1998] Cohen, P., Schrag, R., Jones, E., Pease, A., Lin, A., Starr, B., Gunning, D., and Burke, M. (1998). The DARPA high-performance knowledge bases project[5]. *AI Magazine*, 19(4):25–49.

[Costello, 1997] Costello, T. (1997). *Non-monotonicity and Change*[6]. PhD thesis, Stanford University.

[Costello and McCarthy, 1999] Costello, T. and McCarthy, J. (1999). Useful Counterfactuals[7]. *Electronic Transactions on Artificial Intelligence*.

[Cyc, 2003] Cyc (2003). Cycorp: Makers of the Cyc Knowledge Server for Artificial Intelligence-Based Common Sense[8]. Webpage.

[da Silva, 2003] da Silva, P. P. (2003). Personal Communication.

[4] http://www4.ncsu.edu:8030/˜rychirko/Papers/conjRef.pdf
[5] http://citeseer.nj.nec.com/cohen98darpa.html
[6] http://www-formal.stanford.edu/tjc/phd.html
[7] http://www-formal.stanford.edu/jmc/counterfactuals.ht ml
[8] http://www.cyc.com/

[Davidson, 1966] Davidson, D. (1966). The logical form of action sentences. In Rescher, N., editor, *The Logic of Decision and Action*, pages 81–95. University of Pittsburgh Press.

[de Kleer, 1986] de Kleer, J. (1986). An assumption-based tms. *Artificial Intelligence*, 28:127–162.

[Dickmann, 1985] Dickmann, M. A. (1985). Larger infinitary languages. In Barwise, J. and Feferman, S., editors, *Model-Theoretic Logics*, pages 317–363. Springer, New York.

[Doherty et al., 1998] Doherty, P., Gustafsson, J., Karlsson, L., and Kvarnström, J. (1998). Temporal Action Logics (TAL): Language Specification and Tutorial[9]. *Electronic Transactions on Artificial Intelligence.*

[Enderton, 1972] Enderton, H. B. (1972). *A Mathematical Introduction to Logic.* Academic Press.

[Etherington et al., 1985] Etherington, D. W., Mercer, R. E., and Reiter, R. (1985). On the adequacy of predicate circumscription for closed-world reasoning. *Computational Intelligence*, 1:11–15.

[Fikes, 2003a] Fikes, R. (2003a). Personal Communication.

[Fikes, 2003b] Fikes, R. (2003b). Personal Communication.

[Fikes, 2003c] Fikes, R. (2003c). Personal communication.

[Fikes et al., 2003] Fikes, R., Jenkins, J., and Zhou, Q. (2003). Including Domain-Specific Reasoners with Reusable Ontologies[10]. In *Proceedings of the 2003 International Conference on Information and Knowledge Engineering (IKE.03).*

[Fikes and Nilsson, 1971] Fikes, R. E. and Nilsson, N. J. (1971). Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208.

[9]http://www.ep.liu.se/ea/cis/1998/015/cis98015-revised.ps
[10]http://ksl.stanford.edu/KSL_Abstracts/KSL-03-05.html

[Friedman, 1978] Friedman, H. (1978). Classically and intuitionistically provably recursive functions. In *Higher Set Theory*, volume 699, pages 21–28. Springer Verlag.

[Friedman and Halpern, 1996] Friedman, N. and Halpern, J. Y. (1996). Belief revision: A critique. In Aiello, L. C., Doyle, J., and Shapiro, S., editors, *KR'96: Principles of Knowledge Representation and Reasoning*, pages 421–431. Morgan Kaufmann, San Francisco, California.

[Gabbay, 1996] Gabbay, D. (1996). Fibred semantics and the weaving of logics: Part I: Modal and intuitionistic logics. *Journal of Symbolic Logic*, 61(4):1057–1120.

[Gabbay, 1992] Gabbay, D. M. (1992). Fibred semantics and the weaving of logics. part 2: Fibring non-monotonic logics. In Csirmaz, L., Gabbay, D. M., and de Rijke, M., editors, *Logic Colloquium '92*, pages 75–94. CSLI Publications.

[Gabbay and Nossum, 1997] Gabbay, D. M. and Nossum, R. T. (1997). Structured contexts with fibred semantics. In *Proceedings of the First International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT-97)*, pages 46 – 55.

[Gärdenfors, 1992] Gärdenfors, P. (1992). Belief revision: An introduction. In Gärdenfors, P., editor, *Belief Revision*, volume 29 of *Cambridge Tracts in Theoretical Computer Science*, pages 1–28. Cambridge University Press, Cambridge, UK.

[Gärdenfors and Makinson, 1988] Gärdenfors, P. and Makinson, D. (1988). Revisions of knowledge systems using epistemic entrenchment. In Vardi, M., editor, *Proceedings of the Second Conference on Theoretical Aspects of Reasoning about Knowledge*, pages 83–95. Morgan Kaufmann Publishers.

[Gelfond and Lifschitz, 1993] Gelfond, M. and Lifschitz, V. (1993). Representing action and change by logic programs. *Journal of Logic Programming*, 17:301–322.

[Gelfond and Przymusinska, 1996] Gelfond, M. and Przymusinska, H. (1996). Towards a theory of elaboration tolerance: logic programming approach[11].

[Genesereth, 2003] Genesereth, M. (2003). Personal Communication.

[Ginsberg and Smith, 1987] Ginsberg, M. L. and Smith, D. E. (1987). Reasoning about action I: a possible worlds approach. In Brown, F. M., editor, *The Frame Problem in Artificial Intelligence: Proceedings of the 1987 Workshop*, pages 233–258. Morgan Kaufmann.

[Giunchiglia and Lifschitz, 1998] Giunchiglia, E. and Lifschitz, V. (1998). An action language based on causal explanation: Preliminary report. In *AAAI/IAAI*, pages 623–630.

[Giunchiglia and Walsh, 1992] Giunchiglia, F. and Walsh, T. (1992). A theory of abstraction. *Artificial Intelligence*, 57(2-3):323–389.

[Green, 1969] Green, C. (1969). Applications of theorem proving to problem solving. In *Proceedings IJCAI 69*, pages 219–240.

[Grice, 1989] Grice, P. (1989). *Studies in the Way of Words*. Harvard University Press.

[Guha, 1991] Guha, R. V. (1991). *Contexts: A Formalization and Some Applications*[12]. PhD thesis, Stanford University. Also published as technical report STAN-CS-91-1399-Thesis, MCC Technical Report Number ACT-CYC-423-91.

[Guha, 2003] Guha, R. V. (2003). Personal Communication.

[Gustafsson and Kvarnström, 2001] Gustafsson, J. and Kvarnström, J. (2001). Elaboration tolerance through object-orientation. In *Working Notes of Common Sense 2001*, pages 134–144. Fifth International Symposium on Logical Formalization of Commonsense Reasoning.

[11] http://www.cs.ttu.edu/~mgelfond/papers/tolerance.ps
[12] http://www-formal.stanford.edu/guha/guha-thesis.ps

[Haas, 1987] Haas, A. R. (1987). The case for domain-specific frame axioms. *Proceedings of the 1987 Workshop on the Frame Problem*, pages 343–348.

[Haugh, 1987] Haugh, B. A. (1987). Simple causal minimizations for temporal persistence and projection. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, pages 218–223.

[Kakas et al., 1999] Kakas, A., Miller, R., and Toni, F. (1999). An Argumentation Framework for Reasoning about Actions and Change[13]. In Gelfond, M., Leone, N., and Pfeifer, G., editors, *Proceedings of the 5th International Conference on Logic Programming and Nonmonotonic Reasoning*, LNAI 1730, pages 78–91. Springer Verlag.

[Katsuno and Mendelzon, 1992] Katsuno, H. and Mendelzon, A. O. (1992). On the difference between updating a knowledge base and revising it. In Gärdenfors, P., editor, *Belief Revision*, pages 183–203. Cambridge University Press.

[Kautz and Selman, 1996] Kautz, H. and Selman, B. (1996). Pushing the envelope: Planning, propositional logic, and stochastic search. In Shrobe, H. and Senator, T., editors, *Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference*, pages 1194–1201, Menlo Park, California. AAAI Press.

[Kautz et al., 1996] Kautz, H. A., McAllester, D., and Selman, B. (1996). Encoding plans in propositional logic. In *Proceedings of the Fifth International Conference on the Principle of Knowledge Representation and Reasoning (KR'96)*, pages 374–384.

[Keller and Wilkins, 1985] Keller, A. M. and Wilkins, M. W. (1985). On the use of an extended relational model to handle changing incomplete information. *IEEE Transactions on Software Engineering*, SE-11(7):620–633.

[Kent, 1978] Kent, W. (1978). *Data and Reality: Basic Assumptions in Data Processing Reconsidered*. North-Holland Publishing Company.

[13] http://www.ucl.ac.uk/~uczcrsm/LanguageE/lpnmr99.ps

[Kolmogorov, 1965] Kolmogorov, A. N. (1965). Three approaches to the quantitative definition of information. *Problems of Information and Transmission*, 1(1):1–7.

[Kraus et al., 1990] Kraus, S., Lehmann, D., and Magidor, M. (1990). Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 44:167–207.

[Lee and Lifschitz, 2001] Lee, J. and Lifschitz, V. (2001). Additive fluents. In *Proceedings of AAAI 2001 Spring Symposium on Answer Set Programming*, pages 116–123. AAAI Press.

[Lehmann, 2001] Lehmann, D. (2001). Nonmonotonic logics and semantics. *Journal of Logic and Computation*, 11(2):229–256.

[Lenat and Guha, 1990] Lenat, D. B. and Guha, R. V. (1990). *Building Large Knowledge-Based Systems: Representation and Inference in the CYC Project*. Addison-Wesley.

[Lewis, 1973] Lewis, D. (1973). *Counterfactuals*. Harvard University Press.

[Lifschitz, 1987] Lifschitz, V. (1987). Formal theories of action. In Brown, F., editor, *The Frame Problem in Artificial Intelligence: Proceedings of the 1987 Workshop*, pages 35–58, Los Altos, CA. Morgan Kaufmann Publishers.

[Lifschitz, 1996] Lifschitz, V. (1996). Two components of an action language. In Buvač, S. and Costello, T., editors, *Working Papers: Common Sense '96*, pages 89–95, Stanford University. Computer Science Department, Stanford University.

[Lifschitz, 2000] Lifschitz, V. (2000). Missionaries and cannibals in the causal calculator. In Cohn, A. G., Giunchiglia, F., and Selman, B., editors, *KR2000: Principles of Knowledge Representation and Reasoning,Proceedings of the Seventh International conference*, pages 85–96. Morgan-Kaufman.

[Maier and Warren, 1982] Maier, D. and Warren, D. S. (1982). Specifying connections for a universal relation scheme database[14]. In *Proceedings of the 1982 ACM*

[14]http://portal.acm.org/citation.cfm?id=582355&coll=portal&dl=ACM&ret=1

SIGMOD International Conference on Management of Data, pages 1–7. ACM Press.

[McCain and Turner, 1997] McCain, N. and Turner, H. (1997). Causal theories of action and change. In Shrobe, H. and Senator, T., editors, *Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference*, pages 460–465, Menlo Park, California. AAAI Press.

[McCarthy, 1959] McCarthy, J. (1959). Programs with Common Sense[15]. In *Mechanisation of Thought Processes, Proceedings of the Symposium of the National Physics Laboratory*, pages 77–84, London, U.K. Her Majesty's Stationery Office. Reprinted in [McCarthy, 1990].

[McCarthy, 1962] McCarthy, J. (1962). Towards a Mathematical Science of Computation[16]. In *Information Processing '62*, pages 21–28. North-Holland. Proceedings of 1962 IFIP Congress.

[McCarthy, 1986] McCarthy, J. (1986). Applications of Circumscription to Formalizing Common Sense Knowledge[17]. *Artificial Intelligence*, 28:89–116. Reprinted in [McCarthy, 1990].

[McCarthy, 1988] McCarthy, J. (1988). Mathematical logic in artificial intelligence. *Daedalus*, 117(1):297–311.

[McCarthy, 1990] McCarthy, J. (1990). *Formalizing Common Sense: Papers by John McCarthy*. Ablex Publishing Corporation, 355 Chestnut Street, Norwood, NJ 07648.

[McCarthy, 1993] McCarthy, J. (1993). Notes on formalizing context. In *IJCAI 93*.

[McCarthy, 1997] McCarthy, J. (1997). Elaboration Tolerance[18]. In *McCarthy's web page*. Dynamic version of [McCarthy, 1998].

[15]http://www-formal.stanford.edu/jmc/mcc59.html
[16]http://www-formal.stanford.edu/jmc/towards.html
[17]http://www-formal.stanford.edu/jmc/applications.html
[18]http://www-formal.stanford.edu/jmc/elaboration.html

[McCarthy, 1998] McCarthy, J. (1998). Elaboration Tolerance[19]. In *Proceedings of the Fourth Symposium on Logical Formalizations of Common Sense Reasoning.*

[McCarthy, 2000] McCarthy, J. (2000). Approximate objects and approximate theories. In Cohn, A. G., Giunchiglia, F., and Selman, B., editors, *KR2000: Principles of Knowledge Representation and Reasoning, Proceedings of the Seventh International conference*, pages 519–26. Morgan-Kaufman.

[McCarthy, 2002a] McCarthy, J. (2002a). Personal communication.

[McCarthy, 2002b] McCarthy, J. (2002b). Personal communication.

[McCarthy, 2003] McCarthy, J. (2003). Personal communication.

[McCarthy and Buvač, 1994] McCarthy, J. and Buvač, S. (1994). Formalizing Context (Expanded Notes). Technical Note STAN-CS-TN-94-13, Stanford University.

[McCarthy and Painter, 1967] McCarthy, J. and Painter, J. (1967). Correctness of a compiler for arithmetic expressions. In Schwartz, J. T., editor, *Proceedings of Symposium in Applied Mathematics, vol 19, Mathematical Aspects of Computer Science*, pages 33–41, Providence, RI. American Mathematical Society.

[McDermott, 1978] McDermott, D. (1978). Tarskian semantics, or no notation without denotation! *Cognitive Science*, 2(3):277–282.

[McDermott, 1987] McDermott, D. (1987). A critique of pure reason. *Journal of Computational Intelligence*, 3(3):151–160.

[McGuinness et al., 2000] McGuinness, D. L., Fikes, R., Rice, J., and Wilder, S. (2000). An Environment for Merging and Testing Large Ontologies[20]. In *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR2000); Breckenridge, Colorado.*

[McIlraith, 2000] McIlraith, S. (2000). An axiomatic solution to the ramification problem. *Artificial Intelligence*, 116:87–121.

[19]http://www-formal.stanford.edu/jmc/elaboration.html
[20]http://www.ksl.stanford.edu/KSL_Abstracts/KSL-00-16.html

[McIlraith, 2003] McIlraith, S. (2003). Personal Communication.

[Morgenstern, 1998] Morgenstern, L. (1998). Common Sense Problem Page[21]. Webpage.

[Moulin, 1985] Moulin, H. (1985). Choice functions over a finite set: A summary. *Social Choice and Welfare*, 2:147–160.

[Parmar, 2002] Parmar, A. (2002). Formalizing elaboration tolerance. Thesis Proposal.

[Pearl, 1988] Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.

[Pearl and Verma, 1987] Pearl, J. and Verma, T. (1987). The logic of representing dependencies by directed graphs. In *Proceedings of the 6th National Conference on AI (AAAI-87)*, volume 2, pages 374–379.

[Perez and Jiroušek, 1985] Perez, A. and Jiroušek, R. (1985). Constructing an intensional expert system (ines). In van Bemmel, J., Grémy, F., and Zvárová, J., editors, *Medical Decision Making: Diagnostic Strategies and Expert Systems*, pages 307–315. North-Holland, Amsterdam.

[Reiter, 1980] Reiter, R. (1980). A logic for default reasoning. *Artificial Intelligence*, 13 (1–2):81–132.

[Reiter, 1991] Reiter, R. (1991). The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression[22]. In Lifschitz, V., editor, *Artificial Intelligence and Mathematical Theory of Computation*, pages 359–380. Academic Press.

[Schubert, 1990] Schubert, L. (1990). Monotonic solution of the frame problem in the situation calculus: An efficient method for worlds with fully specified actions. In Kyburg, H. E., Loui, R. P., and Carlson, G. N., editors, *Knowledge Representation*

[21] http://www-formal.stanford.edu/leora/cs/

[22] http://www.cs.toronto.edu/cogrobo/simple.ps.Z

and Defeasible Reasoning, volume 5, pages 23–67. Kluwer Academic Publishers, Dordrecht / Boston / London.

[Sejnowski and Rosenberg, 1986] Sejnowski, T. J. and Rosenberg, C. R. (1986). Nettalk: a parallel network that learns to read aloud. Technical report, Johns Hopkins University. Electrical Engineering and Computer Science Dept.

[Sejnowski and Rosenberg, 1988] Sejnowski, T. J. and Rosenberg, C. R. (1988). Nettalk: a parallel network that learns to read aloud. In Anderson, J. A. and Rosenfeld, E., editors, *Neurocomputing*, volume Volume 1: Foundations of Research, pages 663–672. MIT Press, Cambridge, MA. Paper originally published as [Sejnowski and Rosenberg, 1986].

[Semantic Web, 2002] Semantic Web (2002). Semantic web.

[Sen, 1970] Sen, A. K. (1970). *Collective Choice and Social Welfare*. Holden-Day, San Francisco.

[Shanahan, 1997] Shanahan, M. (1997). *Solving the Frame Problem: A Mathematical Investigation of the Common Sense Law of Inertia*. M.I.T. Press.

[Simons, 2000] Simons, P. (2000). Smodels Web Page[23]. Web page.

[Subramanian and Genesereth, 1987] Subramanian, D. and Genesereth, M. R. (1987). The relevance of irrelevance. In *Proceedings of IJCAI-87*, volume 1, pages 416 – 422.

[Syrjänen, 2000] Syrjänen, T. (2000). *Lparse User's Manual (Draft 1.0)*[24]. Digital Systems Laboratory, Helsinki University of Technology.

[van Benthem, 1988] van Benthem, J. (1988). *A Manual of Intensional Logic*. CSLI Publications.

[van Benthem, 2003] van Benthem, J. (2003). Personal communication.

[23]http://www.tcs.hut.fi/Software/smodels/
[24]http://www.tcs.hut.fi/Software/smodels/lparse/lparse.ps.gz

[van Eijck and Kamp, 1997] van Eijck, J. and Kamp, H. (1997). Representing discourse in context. In van Benthem, J. and ter Meulen, A., editors, *Handbook of Logic and Language*, pages 180–237. North-Holland.

[Winslett, 1989] Winslett, M. (1989). Sometimes updates are circumscription. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI'89)*, pages 859–863.