Toward a Formalization of Elaboration Tolerance: Adding and Deleting Axioms

Eyal Amir
Department of Computer Science, Gates 2A
Stanford University, Stanford, CA 94305-9020, USA
eyala@cs.stanford.edu
http://www-formal.stanford.edu/eyal

Abstract

When creating a knowledge base, a knowledge engineer faces design and modeling choices. The decisions taken may later affect the evolution of the knowledge base. Indeed, when new knowledge needs to be integrated, the knowledge base may have to undergo some rewriting and redesign, in order to incorporate the new information. Creating the Knowledge Base such that future changes are made easier is a major concern of the knowledge engineer. We refer to the ease of change as *Elaboration Tolerance*.

In order to implement and evaluate Elaboration Tolerance in formal systems, it is necessary to give a formal definition and to provide evaluation tools. These definition and tools constitute the major contribution of this paper. We propose a formal definition for one aspect of the problem of Elaboration and supply tools for comparing Knowledge Bases with respect to their Elaboration Tolerance. These definitions approximate the problem of Elaboration from below. We supply examples illustrating the intuitions captured by these definitions and tools. We then demonstrate the use of these definitions and tools by applying them to examples of language expansion and monotonic versus nonmonotonic reasoning, and examine the limits of the approach.

1 Introduction

The notion of *Elaboration Tolerance* was proposed by McCarthy (McCarthy 1988) for the problem of extending a logical theory, with the intuition that a logical system should have the ability to absorb additions the way Natural Language allows. Several intuitions coincide in this description: axiomatizing a theory in a flexible way; not needing to rebuild one's ontology when new features and facts are added; Being able to modify one's axioms relatively easily; and needing only a small amount of recomputation, given new information. Elaboration Tolerance is important for Knowledge Base construction and development, as well as for scaling up results and techniques in Knowledge Representation. Past treatments of the concept referred only to intuitive accounts. In order to investigate Elaboration Tolerance, however, we need formal definitions and comparison tools. The major contribution of this work is supplying such tools.

In this paper, we propose a formal definition for one *aspect* of the property of Elaboration Tolerance. We refer to Elaborations as sequences of actions as executed on a certain Knowledge Base. In this paper, we restrict our treatment to Knowledge Bases that are formal systems and to actions that add or delete axioms (other possible actions that we ignore here are adding constants to the language, adding preconditions to axioms, etc., depending on what is allowed by the knowledge base). We intuitively say that a Knowledge Base is *elaboration-tolerant* to the extent that elaborations are representable and require "short" sequences of actions for these elaborations. The *Problem of Elaboration* (as examined in this paper) is, given an intended elaboration (e.g., using semantics to specify these intentions), finding a sequence of actions that will give us the intended result.

To keep our discussion simple, we restrict our treatment in this paper in two ways: we allow the addition and deletion of disjunctive clauses only, and we restrict the formal systems to have a propositional language. The reason for the first restriction will become clear in section 2.3. The second restriction simplifies most of the theorems but most of them follow for the general First-Order case.

Using this framework, we show the following (some intuitive and some surprising) results. On the intuitive side, we show that a propositional knowledge base with a larger set of constant symbols is more elaboration-tolerant than an equivalent one with fewer symbols. On a somewhat less intuitive angle, we show that some nonmonotonic theories are not more elaboration-tolerant than some equivalent monotonic theories. We then show that, despite this fact, from a monotonic theory there is a way to construct an equivalent nonmonotonic theory that is more elaboration-tolerant than the original one. Finally, we show that there is no one most elaboration-tolerant system.

A few authors have informally discussed Elaboration Tolerance in the past. (McCarthy 1998) gives some examples and discusses intuitions of Elaboration Tolerance. It attributes the creation of Nonmonotonic Reasoning techniques to Elaboration Tolerance. (Shanahan 1996) further discusses Elaboration Tolerance and shows how the desire for Elaboration Tolerance invigorates major portions of the Knowledge Representation endeavor. (Costello 1997) and (Amir 1997) showed how different theories of action relate with respect to certain elaborations. Other relevant material is (Giunchiglia & Walsh 1992) and the work on abstraction. This work is relevant in supplying both elaborations (reverse abstractions[1]) and translation functions (used in comparing systems).

In contrast to these discussions, we try to define a region on which some formal work can be done. We seek guidelines for choosing a formal machinery and writing the theory so that extensions are simple. The theory that we provide in this paper aids in finding these guidelines by focusing on the number of actions that need to be performed in order to amend the knowledge base. Consequently, we wish to minimize the length of that sequence of actions.

2 The Elaboration Model

This section is the core of the paper. Here we will first go through an intuitive explanation, then define our notions and eventually explore a detailed example using these notions.

2.1 Intuitions & Intentions

We want to compare knowledge base design decisions, preferring those decisions that give us more Elaboration Tolerance. One simple intuitive example is the following. Assume that we are given the logical theory

$$Rain \implies Cold \tag{1}$$
$$Wet \implies Cold \tag{2}$$

Now, we want to add the fact that in the tropics it is not cold but it may rain and it is wet. One way to do that is to rewrite the theory from scratch:

$$Rain \implies (Cold \lor Tropics)$$
$$Wet \implies (Cold \lor Tropics)$$

Consider the following theory, which is similar to the original theory above.

$$Rain \implies Preconditions \tag{3}$$
$$Wet \implies Preconditions \tag{4}$$
$$Preconditions \implies Cold \tag{5}$$

In this theory we added the observation that being wet and having rain are two properties that share preconditions. In this case, instead of rewriting the entire theory from scratch, we can simply replace the third sentence (5) with

$$Preconditions \implies (Cold \lor Tropics)$$

We would like to prefer compact changes, like the last one, over rewriting extended sections of our knowledge base, as seen in the first case.

[1]A proposal made by Alon Levy in a personal communication.

Coming up with a complete characterization of the set of possible tricks one might pull seems like an open-ended task. These "tricks" may include design decisions like Object-Oriented design (e.g., using Frames (see (Minsky 1975) and (Brachman & Levesque 1985))), but may also include simple aggregations such as the one demonstrated in the example just described.

To understand the tradeoffs and to find new ways of enhancing the Elaboration Tolerance of a knowledge base, we now turn to introduce a formal account of the comparisons that we wish to make and the qualifications that we want to measure.

2.2 A Model of Knowledge-Base Change

In this section we describe a model for knowledge-base change. We take a syntactic approach to knowledge-base change since, as the example above demonstrated, the actions done by a human knowledge engineer are syntactic (for relations to Belief Revision, see section 5). For now, we ignore the decisions that the knowledge engineer may face and their complexity, and focus on the final product of her changes.

We restrict our discussion to knowledge bases that are *Axiomatic Formal Systems.* In later sections we will see the reason we need such a broad definition.

Definition 2.1 (From (Shoenfield 1967)) *An* Axiomatic Formal System Σ *is a triple* $\langle \mathcal{L}, \mid\!\sim, \Gamma \rangle$ *where* $\mathcal{L}$ *is the language,* Γ *is the set of axioms and* $\mid\!\sim$ *is the inference relation*[2] *of* Σ.

In this paper, we treat the language as its set of sentences and we use the notation $|\mathcal{L}|$ to represent the set of propositional symbols of the language $\mathcal{L}$. For $\Sigma = \langle \mathcal{L}, \mid\!\sim, \Gamma \rangle$, we take $C(\Sigma)$ to be the theory entailed by the formal system's axioms (C comes for the *completion* of Σ):

$$C(\Sigma) \stackrel{def}{=} \{\varphi \in \mathcal{L} \mid \Gamma \mid\!\sim \varphi\}$$

We define an equivalence relation between formal systems as follows:

$$\Sigma \equiv \Sigma' \iff C(\Sigma) = C(\Sigma').$$

Notice that $C(\Sigma) = C(\Sigma')$ is determined extensionally (equality of sets). Also, notice that although this definition allows different languages in Σ and Σ', the equality of languages is in fact implied for all logics that entail tautologies.

In what follows we describe actions for changing the knowledge base. In fact, each action transforms a formal system into another formal system. In this paper we restrict our attention to adding and deleting axioms.

If $\varphi \subset \mathcal{L}$ is an *additional axiom*, then we write $add(\varphi)(\Sigma)$ for the result of adding φ to Σ, i.e.,

$$add(\varphi)(\Sigma) \stackrel{def}{=} \langle \mathcal{L}, \mid\!\sim, \Gamma \cup \{\varphi\} \rangle.$$

[2]Possibly nonmonotonic.

To *delete an axiom* from the knowledge base, we use a similar action $delete(\varphi)(\Sigma)$:

$$delete(\varphi)(\Sigma) \stackrel{def}{=} \langle \mathcal{L}, \mid\!\sim, \Gamma \setminus \{\varphi\} \rangle.$$

Notice that because of the syntactic nature of these actions, deleting an axiom that does not exist in the knowledge base (even though it may be entailed by the knowledge base) results in the original knowledge base (we prefer this definition over leaving it undefined).

To represent the result of a sequence of actions performed on the knowledge base, we shall use the convention that applying the sequence of actions $\alpha = \langle a_1, ..., a_n \rangle$ (where each of the actions $\{a_i\}_{i \leq n}$ is of the form $add(\varphi)$ or $delete(\varphi)$ for a sentence φ in the relevant language) is written as

$$\alpha(\Sigma) \stackrel{def}{=} a_n(a_{n-1}(...(a_1(\Sigma))...)).$$

Thus, if $\alpha = \langle delete(\psi), add(\varphi) \rangle$ then $\alpha(\Sigma) = add(\varphi)(delete(\psi)(\Sigma))$. Finally, we write $Actions(\mathcal{L})$ for all the sequences α of actions that can be performed in $\mathcal{L}$.

The reason we chose to focus on the actions of adding and deleting axioms is two-fold: first, these action seem to be the simplest, yet extremely common, operations done during knowledge base construction and expansion, and second, in this stage of inquiry, constraining ourselves to only two, very simple, actions, helps in clarifying and focusing the investigation and explanation.

2.3 Comparing Elaboration Tolerance

Now we turn to define a simplified account of Elaboration Tolerance. Intuitively, we ignore the difficulties encountered in formalizing the process of conceptualization, and focus only on the syntactic work that has to be done (possibly after long deliberation) to change the knowledge base to its target meaning.

To compare two systems, we first have to translate them to some "common ground". Somewhat similar to the approach taken by (Giunchiglia & Walsh 1992), we first define a translation between two formal systems ((Giunchiglia & Walsh 1992) call their function an *abstraction function*, but this name is not necessarily appropriate in our context).

Definition 2.2 (Translation) *A* translation *t is a partial function that accepts a formal system and returns a formal system. In other words, $t : S \rightarrow S$ a partial function, where S is a set of axiomatic formal systems.*

We can now use the equivalence relation defined by t:

$$DEF: \quad \Sigma_1 \equiv_t \Sigma_2 \iff C(t(\Sigma_1)) = C(t(\Sigma_2))$$

Notice that this is indeed an equivalence relation. Also, notice that the translation t does not translate Σ_1 to Σ_2 but rather translates both formal system to one (possibly different than both) common ground.

Now we define the problem of elaboration to be the problem of finding a sequence of actions that transforms an original knowledge base to a desired knowledge base (modulo our equivalence relation $\equiv_t$).

Definition 2.3 (Elaboration Problem) *We are given two axiomatic formal systems, Σ, Σ_{target} and a translation t. The* problem of elaboration *is to find an elaboration (i.e., a sequence of actions) α such that $\alpha(\Sigma) \equiv_t \Sigma_{target}$.*

Given this definition, we would like to give a criterion for Σ' to be more elaboration-tolerant than Σ with respect to the target Σ_{target}. The following measure tries to capture the difficulty encountered in the syntactic operations performed on the knowledge base.

Definition 2.4 (Elaboration Distance) *Let Σ, Σ_{target} be two axiomatic formal systems. The* Elaboration Distance[3] *of Σ_{target} from Σ is*

$$dist_t(\Sigma, \Sigma_{target}) \stackrel{\text{def}}{=} \min \left\{ len(\alpha) \;\middle|\; \begin{array}{l} \alpha \in Actions(\mathcal{L}) \\ \wedge\ \alpha(\Sigma) \equiv_t \Sigma_{target} \end{array} \right\}$$

where $len(\alpha)$ is the number of actions in α. We take $dist_t(\Sigma, \Sigma_{target})$ to be ∞ in the case that there is no sequence of actions that will transform Σ to Σ_{target}.

Intuitively, we wish $dist_t(\Sigma, \Sigma_{target})$ to measure the minimal work we need to do in order to change Σ into something equivalent (modulo $\equiv_t$) to Σ_{target}. Elaboration Distance (in the case where there is an elaboration (i.e., $dist_t(\Sigma, \Sigma') < \infty$)) is useful when we want to compare two systems that can both represent the same elaborations. Recalling the intuition that we wish to minimize the work done in changing the knowledge base, the weight function ($len(\alpha)$) that we use here does not seem convincing. Why does this $dist_t$ capture our intuition of *difficulty* of expanding the theory?

First, it is reasonable to say that the less you have to add to your theory, the more robust it was to begin with, so this measure approximates some of our intuitions of Elaboration Tolerance. The real problem in choosing a weight function for this case is that many people have different intuitions for it. One would rather have it be either 0 (there *is* an elaboration) or ∞ (there *is no* such elaboration). Another would rather have the weight measure the *difficulty of finding the elaboration* (however that might be formulated). I propose this function as a quick-and-dirty measure that one can do with for some time. Most of the results (e.g., theorems 4.1 and 4.2) are independent of which measure we take (given that, if there is no elaboration, we get ∞).

To compare two systems Σ_1 and Σ_2, we use the translation t to give Σ_1 and Σ_2 a common ground. It is important to notice that the comparison is highly dependent on our choice of t (one possible choice of t is demonstrated in section 4).

[3] This is a quasi-distance measure, i.e., it is not symmetric.

Definition 2.5 (Elaboration Comparison) *Let $\Sigma_1 = \langle \mathcal{L}_1, \mid\!\sim_1, \Gamma_1 \rangle$, $\Sigma_2 = \langle \mathcal{L}_2, \mid\!\sim_2, \Gamma_2 \rangle$ be two axiomatic formal systems and t a translation such that $\Sigma_1 \equiv_t \Sigma_2$.*

Given t, define Σ_1 to be more elaboration-tolerant *than Σ_2 on $\alpha \in Actions(\mathcal{L}_1)$, by*

$$\Sigma_1 \leq_{t,\alpha} \Sigma_2 \stackrel{\text{def}}{=} dist_t(\Sigma_1, \alpha(\Sigma_1)) \leq dist_t(\Sigma_2, \alpha(\Sigma_1)).$$

Given t, define Σ_1 to be more elaboration-tolerant *than Σ_2, by*

$$\Sigma_1 \leq_t \Sigma_2 \stackrel{\text{def}}{=} \forall \alpha \in Actions(\mathcal{L}_1)\ \Sigma_1 \leq_{t,\alpha} \Sigma_2 \wedge \forall \alpha \in Actions(\mathcal{L}_2)\ \exists \alpha' \in Actions(\mathcal{L}_1)\ (\alpha'(\Sigma_1) \equiv_t \alpha(\Sigma_2))$$

For the strict cases, we require that there is an elaboration $\alpha \in Actions(\mathcal{L}_1)$ such that $dist_t(\Sigma_1, \alpha(\Sigma_1)) < dist_t(\Sigma_2, \alpha(\Sigma_1))$. For that case, we say *strictly more elaboration-tolerant* and write the corresponding strict inequality sign.

We postpone the re-examination of the example covered in beginning of this section to section 4.2, where we prove a general theorem that captures this example.

2.4 Examples

Let us look at a variant of the example given in section 2.1. Assume that we have a theory describing some aspects of weather. For simplicity, assume that the theory has the following axioms:

$$Rain \Longrightarrow Preconditions \qquad (6)$$

$$Preconditions \Longrightarrow Cold \qquad (7)$$

Let $\Sigma = \langle \mathcal{L}, \vdash, \{(6), (7)\} \rangle$ where $\mathcal{L}$ is the propositional language of the propositional symbols $Rain, Preconditions, Cold, Tropics$ and $\vdash$ is the classical propositional entailment relation.

Now we wish to say that it is possible that there will be rain without being cold if we are in the tropics. We will have to perform the sequence of actions $\alpha = \langle delete((7)), add(\text{“}Preconditions \Longrightarrow (Cold \vee Tropics)\text{''}) \rangle$. The resulting formal system $\alpha(\Sigma)$ then has the same language $\mathcal{L}$, the same entailment relation $\vdash$ and the following set of axioms:

$$\begin{array}{l} Rain \Longrightarrow Preconditions \\ Preconditions \Longrightarrow (Cold \vee Tropics) \end{array} \qquad (8)$$

Thus, $\alpha(\Sigma) = \Sigma_{target} = \langle \mathcal{L}, \vdash, \{(8)\} \rangle$. Because of the monotonicity of propositional logic, this change cannot be done if we allow only the addition axioms (without deletion), and thus there is no way to perform the change with only one action.

There are other systems that will allow for that elaboration in a shorter manner. Take, for example, the following nonmonotonic system. Assume that, instead of the original set of axioms, we have

$$\begin{array}{l} Rain \Longrightarrow Preconditions \\ Cold \Longrightarrow Preconditions \end{array} \qquad (9)$$

The system is $\Sigma' = \langle \mathcal{L}, \mid\!\sim_C, \{(9)\} \rangle$, where

$$\Gamma \mid\!\sim_C \varphi \iff Circ[\Gamma; Preconditions; \mathcal{L}] \vdash \varphi$$

Here, $Circ[A; P; Q]$ is the circumscription formula $A(P, Q) \wedge \forall pq[A(p, q) \Rightarrow \neg(p < P)]$, as defined in (McCarthy 1980), which intuitively says that P is minimized even at the price of changing Q. In our context, that means (intuitively) that the proposition $Preconditions$ take the truth value FALSE, unless it “must” be TRUE. We can get the result required by Σ_{target} (our target system above) by simply adding sentence

$$Tropics \Longrightarrow Preconditions \qquad (10)$$

Basically, what we are saying is that every formula is assumed to be true, given no contradictory information. Without the knowledge (10), a result of minimizing the preconditions would be that $Preconditions \iff Cold$. Given the additional knowledge (10), we conclude that $Preconditions \iff (Cold \vee Tropics)$.

If we take t to be the identity translation $t(\Sigma) = \Sigma$, then we can conclude $\Sigma \equiv_t \Sigma'$ and that

$$\Sigma' \leq_{t,\alpha} \Sigma$$

for $\alpha = \langle add(\text{“}Tropics \Longrightarrow Preconditions\text{''}) \rangle$. Notice, that α itself can be applied to Σ, but it will yield a different result than Σ_{target}.

3 Some Intermediate Results

We prove a few basic properties of the above relations. Notice that we put no restriction on t.

Proposition 3.1 *If $\Sigma_1 \leq_{t,\alpha_1} \Sigma_2$ and $\Sigma_2 \leq_{t,\alpha_2} \Sigma_3$ and $\alpha_1(\Sigma_1) \equiv_t \alpha_2(\Sigma_2)$, then $\Sigma_1 \leq_{t,\alpha_1} \Sigma_3$. For the strict case, if any of the two preconditions is a strict inequality, the result is also a strict inequality.*

Proof By the definition, we get $dist_t(\Sigma_1, \alpha_1(\Sigma_1)) \leq dist_t(\Sigma_2, \alpha_1(\Sigma_1))$ and $dist_t(\Sigma_2, \alpha_2(\Sigma_2)) \leq dist_t(\Sigma_3, \alpha_2(\Sigma_2))$. Since $\alpha_1(\Sigma_1) \equiv_t \alpha_2(\Sigma_2)$, the first inequality implies that $dist_t(\Sigma_1, \alpha_1(\Sigma_1)) \leq dist_t(\Sigma_2, \alpha_2(\Sigma_2))$ and so $dist_t(\Sigma_1, \alpha_1(\Sigma_1)) \leq dist_t(\Sigma_3, \alpha_2(\Sigma_2))$ and finally $dist_t(\Sigma_1, \alpha_1(\Sigma_1)) \leq dist_t(\Sigma_3, \alpha_1(\Sigma_1))$.

For the strict cases, assume first that $\Sigma_1 <_{t,\alpha_1} \Sigma_2$. By the definition, we get $dist_t(\Sigma_1, \alpha_1(\Sigma_1)) < dist_t(\Sigma_2, \alpha_1(\Sigma_1))$ and $dist_t(\Sigma_2, \alpha_2(\Sigma_2)) \leq dist_t(\Sigma_3, \alpha_2(\Sigma_2))$. Since $\alpha_1(\Sigma_1) \equiv_t \alpha_2(\Sigma_2)$, we get that $dist_t(\Sigma_1, \alpha_1(\Sigma_1)) < dist_t(\Sigma_2, \alpha_2(\Sigma_2))$ and so $dist_t(\Sigma_1, \alpha_1(\Sigma_1)) < dist_t(\Sigma_3, \alpha_2(\Sigma_2))$ and finally $dist_t(\Sigma_1, \alpha_1(\Sigma_1)) < dist_t(\Sigma_3, \alpha_1(\Sigma_1))$. The other strict case is identically treated. ■

Corollary 3.2 *If $\Sigma_1 \leq_t \Sigma_2$ and $\Sigma_2 \leq_t \Sigma_3$, then $\Sigma_1 \leq_t \Sigma_3$.*

Proof Let $\alpha_1 \in Actions(\mathcal{L}_1)$. If there is $\alpha_2 \in Actions(\mathcal{L}_2)$ such that $\alpha_1(\Sigma_1) \equiv_t \alpha_2(\Sigma_2)$, then by proposition 3.1 and from $\Sigma_1 \leq_{t,\alpha_1} \Sigma_2$ and $\Sigma_2 \leq_{t,\alpha_2} \Sigma_3$, we get that $\Sigma_1 \leq_{t,\alpha_1} \Sigma_3$.

If there is no such α_2 but there is $\alpha_3 \in Actions(\mathcal{L}_3)$ such that $\alpha_3(\Sigma_3) \equiv_t \alpha_1(\Sigma_1)$, then, because of the *second requirement* of $\leq_t$, there is $\alpha_2(\Sigma_2) \equiv_t \alpha_3(\Sigma_3)$. Thus, $\alpha_2(\Sigma_2) \equiv_t \alpha_1(\Sigma_1)$. Contradiction. Therefore, there is no such α_3, and $\Sigma_1 \leq_{t,\alpha_1} \Sigma_3$.

To prove the second requirement of $\Sigma_1 \leq_t \Sigma_3$, notice that if $\alpha_3 \in Actions(\mathcal{L}_3)$, then $\exists\alpha_2 \in Actions(\mathcal{L}_2)$ such that $\alpha_3(\Sigma_3) \equiv_t \alpha_2(\Sigma_2)$ and $\exists\alpha_1 \in Actions(\mathcal{L}_1)$ such that $\alpha_2(\Sigma_2) \equiv_t \alpha_1(\Sigma_1)$. ■

Corollary 3.3 *$\leq_t$ is a pre-order. $<_t$ is a strict partial order.*

PROOF The reflexivity for $\leq_t$ is obvious, and thus $\leq_t$ is a pre-order.

For the irreflexivity of $<_t$, assume that $\Sigma <_{t,\alpha} \Sigma$ for $\alpha \in Actions(\mathcal{L})$. Then $dist_t(\Sigma, \alpha(\Sigma)) < dist_t(\Sigma, \alpha(\Sigma))$. Contradiction. Thus, we have irreflexivity.

For transitivity of $<_t$, assume $\Sigma_1 <_t \Sigma_2 <_t \Sigma_3$. Then, $\exists\alpha_1 \Sigma_1 <_{t,\alpha_1} \Sigma_2$. If there is no $\alpha_2 \in Actions(\mathcal{L}_2)$ such that $\alpha_1(\Sigma_1) \equiv_t \alpha_2(\Sigma_2)$, then there is no $\alpha_3 \in Actions(\mathcal{L}_3)$ such that $\alpha_1(\Sigma_1) \equiv_t \alpha_3(\Sigma_3)$ and $\Sigma_1 <_{t,\alpha_1} \Sigma_3$. If there is such α_2, then $\Sigma_2 \leq_{t,\alpha_2} \Sigma_3$ and by proposition 3.1 (the strict case), $\Sigma_1 <_{t,\alpha_1} \Sigma_3$. ■

The following lemma will become useful in the next section.

Lemma 3.4 *Let Σ_1, Σ_2 be formal systems such that $\Sigma_1 \equiv_t \Sigma_2$. Assume $\mathcal{L}_2 \subseteq \mathcal{L}_1$ and $\forall\alpha \in Actions(\mathcal{L}_2)$ $\alpha(\Sigma_1) \equiv_t \alpha(\Sigma_2)$. Then, $\Sigma_1 \leq_t \Sigma_2$.*

PROOF Take $\alpha \in Actions(\mathcal{L}_1)$. We need to show $\Sigma_1 \leq_{t,\alpha} \Sigma_2$. Let $\alpha' \in Actions(\mathcal{L}_2)$ such that $\alpha(\Sigma_1) \equiv_t \alpha'(\Sigma_2)$. Since $\alpha' \in Actions(\mathcal{L}_1)$ and $\alpha'(\Sigma_2) \equiv_t \alpha'(\Sigma_1)$ (by the lemma's conditions), $\alpha'(\Sigma_1) \equiv_t \alpha(\Sigma_1)$. Thus, $dist_t(\Sigma_1, \alpha(\Sigma_1)) \leq dist_t(\Sigma_2, \alpha(\Sigma_1))$, and since α was arbitrary, the first condition is proved. The second requirement is supplied by the premises of the lemma. ■

4 Comparisons Using *Consequence Translations*

In this section, we describe various knowledge bases and compare their elaboration tolerance using translations t that are *consequence translations*. Recall that a translation is used to compare knowledge bases by mapping both to some common-ground on which they agree.

The *consequence translation* to $\mathcal{L}, \mid\!\sim$ is $t(\langle\mathcal{L}_0, \mid\!\sim_0, \Gamma_0\rangle) = \langle\mathcal{L}, \mid\!\sim, \Gamma\rangle$ for $\Gamma = C(\langle\mathcal{L}_0, \mid\!\sim_0, \Gamma_0\rangle) \cap \mathcal{L}$ ((Giunchiglia & Walsh 1992) have somewhat similar mappings they call *Predicate Abstractions* and *AB-STRIPS* abstractions). Intuitively, a consequence translation maps a formal system to a new formal system that has a preset language and a preset entailment relations (both "preset" for that consequence translation). The set of axioms of the new formal system is exactly the set of consequences of the axioms of the original formal system (using the original entailment relation).

Throughout this section we give special treatment to the case that only adding-axioms actions are allowed. This special case turns out to have some nice intuitive properties (some of the theorems proved below are true only for that case) and contrasting these properties with those of the general case is instructive.

4.1 Different Languages

Next, we show that adding a constant symbol to a propositional theory increases the Elaboration Tolerance.

Let us look at the following theory.

$$Rain \implies Clouds \tag{11}$$

$$Clouds \implies \neg Sun \tag{12}$$

the language includes the propositional symbols $Rain, Clouds, Sun$ and the entailment relation is the propositional entailment relation $\vdash$. Assume now that we added the propositional symbol $Moon$ to our language. Every elaboration that we could do before is still a valid elaboration here. More importantly, it yields the same results that it had in the original system (see the proof of the theorem below). Aside from that, there are some new elaborations that we can include, such as adding the axiom $Moon \iff \neg Sun$.

Theorem 4.1 *Let $\vdash$ be the propositional entailment relation. Let $\mathcal{L}_2 \subseteq \mathcal{L}_1$ be two propositional languages and t be the consequence translation to $\mathcal{L}_2, \vdash$. If $\Sigma_1 = \langle\mathcal{L}_1, \vdash, \Gamma_1\rangle$, $\Sigma_2 = \langle\mathcal{L}_2, \vdash, \Gamma_2\rangle$ such that $\Sigma_2 \equiv_t \Sigma_1$ and only axiom-adding actions are allowed (for both systems), then $\Sigma_1 \leq_t \Sigma_2$ (Σ_1 is more elaboration-tolerant than Σ_2).*

PROOF Let $\alpha_2 \in Actions(\mathcal{L}_2)$. We want to prove that $\alpha_2(\Sigma_2) \equiv_t \alpha_2(\Sigma_1)$, which will allow us to use lemma 3.4 to complete the proof.

$\Sigma_1 \equiv_t \Sigma_2$ implies that $\mathcal{L}_2 \cap C(\Sigma_1) = C(\Sigma_2)$. α_2 adds a sentence (or a conjunction of sentences), $\varphi \in \mathcal{L}_2$. Since $\varphi \in \mathcal{L}_2$, $\mathcal{L}_2 \cap (C(\Sigma_1) \cup \{\varphi\}) \equiv \mathcal{L}_2 \cap C(\Sigma_1 \wedge \varphi)$. This is because we can replace every axiom $\xi \in C(\Sigma_1 \wedge \varphi) \cap \mathcal{L}_2$, in a proof in the system on the right, with the two axioms $\varphi, \varphi \to \xi$, both in $(C(\Sigma_1) \cup \{\varphi\}) \cap \mathcal{L}_2$. Our ability to do this replacement is a direct result of the deduction theorem.

Since $C(\Sigma_2) \cup \{\varphi\} \equiv C(\Sigma_2 \wedge \varphi)$, we get that $(\mathcal{L}_2 \cap C(\Sigma_1)) \cup \{\varphi\} \equiv C(\Sigma_2) \cup \{\varphi\}$, which implies that $C(\Sigma_2 \wedge \varphi) \equiv C(\Sigma_1 \wedge \varphi) \cap \mathcal{L}_2$. Thus, $\Sigma_2 \wedge \varphi \equiv_t \Sigma_1 \wedge \varphi$ and $\alpha_2(\Sigma_2) \equiv_t \alpha_2(\Sigma_1)$. Now, all that is left is to use lemma 3.4 and get $\Sigma_1 \leq_t \Sigma_2$. ■

Notice that, as a result of this theorem, two logically equivalent propositional formal systems have the same elaboration tolerance (given that we allow only axiom additions and no deletions). Of course, this is true under the assumptions that the translation is a consequence translation to $\mathcal{L}, \vdash$.

If we allow actions that delete axioms then the theorem above is not true. If the theorem held in the general case, we would have concluded that equivalent monotonic systems with the same language have the same elaboration tolerance (allowing both adding and removing axioms). An example that this is not the case was seen in section 2.1.

The proof reveals the pleasant surprise that the last theorem is not in general true for nonmonotonic entailment relations (even if we allow only axiom-adding actions). This is hinted at by the use of the deduction theorem in the proof above and is illustrated by example 4.3 in section 4.2.

4.2 Propositional Monotonic and Nonmonotonic Systems

Our exposition in this section is stated using the nonmonotonic system of Circumscription that we already used in section 2.4. The same results can be stated for the nonmonotonic systems of Default Logic (Reiter 1980), Autoepistemic Logic (Moore 1987) and possibly others, but for simplicity, we restrict ourselves and discuss only the case of Circumscription.

Let $\Sigma_1 = \langle \mathcal{L}_1, \vdash, \Gamma_1 \rangle$, with $\mathcal{L}_1$ a propositional language, $\vdash$ the propositional entailment and Γ_1 a set of axioms in $\mathcal{L}_1$. The *associated abnormality theory* of Σ_1 is $\Sigma_2 = \langle \mathcal{L}_2, \mid\!\sim_C, \Gamma_2 \rangle$, with $\Gamma_2 = \{ab_\varphi \rightarrow \zeta \mid \varphi \in \Gamma_1\}$, $\mathcal{L}_2$ the propositional language $\mathcal{L}_1 \cup \{ab_\varphi | \varphi \in \Gamma_1\}$ (i.e., $\mathcal{L}_2 = \mathcal{L}_1 \cup \mathcal{L}(\Gamma_2)$) and $\mid\!\sim_C$ an entailment relation that first circumscribes the ab_φ's in parallel (varying the propositions of $\mathcal{L}_1$) and then treats the result propositionally. More precisely,

$$\Gamma \mid\!\sim_C \psi \iff Circ[\Gamma; ab_{\varphi_1}, ..., ab_{\varphi_n}; p_1, ..., p_m] \vdash \psi$$

The following theorem says that the associated abnormality theory of a propositional theory is strictly (under some assumptions) more elaboration-tolerant than the original propositional theory:

Theorem 4.2 *Let $\Sigma_1 = \langle \mathcal{L}_1, \vdash, \Gamma_1 \rangle$ and Σ_2 be the associated abnormality theory of Σ_1. Assume that Σ_1 is not a contradiction (If it is a contradiction, then $\Sigma_2 \not\equiv_t \Sigma_1$) and assume that we are allowed only the addition of axioms (no axiom-deleting actions). Then $\Sigma_2 \leq_t \Sigma_1$, for t the consequence translation to $\mathcal{L}_1, \vdash$. If, in addition, Σ_1 is not a tautology, then $\Sigma_2 <_t \Sigma_1$.*

PROOF First, notice that $\Sigma_2 \equiv_t \Sigma_1$. It is also simple to see that for all $\alpha_1 \in Actions(\mathcal{L}_1)$, that does not cause inconsistency (i.e., $\alpha_1(\Sigma_1)$ is consistent), $\alpha_1(\Sigma_2) \equiv_t \alpha_1(\Sigma_1)$. The reason is that, if the axioms are consistent, then the circumscription will simply entail that all the ab's are false, leading to an equivalent theory $\mathcal{L}_1 \cap \alpha_1(\Sigma_2) \equiv \alpha_1(\Sigma_1)$.

In the case of inconsistency in $\alpha_1(\Sigma_1)$, take[4] $\alpha_2 = \{\{\}\}$. α_2 causes inconsistency in Σ_2. Thus, we have shown the second condition of $\Sigma_2 \leq_t \Sigma_1$.

To show the first condition of $\Sigma_2 \leq_t \Sigma_1$, take $\alpha \in Actions(\mathcal{L}_2)$. We need to show that $dist_t(\Sigma_2, \alpha(\Sigma_2)) \leq dist_t(\Sigma_1, \alpha(\Sigma_2))$. Assume that this is not the case. Then there is $\alpha' \in Actions(\mathcal{L}_1)$ such that $\alpha(\Sigma_2) \equiv_t \alpha'(\Sigma_1)$ and for all $\beta \in Actions(\mathcal{L}_2)$ such that $\alpha(\Sigma_2) \equiv_t \beta(\Sigma_2)$, $len(\alpha') < len(\beta)$. If $\alpha'(\Sigma_1)$ is not inconsistent, then $\alpha' \in Actions(\mathcal{L}_2)$ and $\alpha'(\Sigma_2) \equiv_t \alpha'(\Sigma_1)$, contradicting our claim about α'. If $\alpha'(\Sigma_1)$ is inconsistent, then $\alpha(\Sigma_2)$ is inconsistent, thus having $\beta = \{\{\}\}$ with $\alpha(\Sigma_2) \equiv_t \beta(\Sigma_2)$ and again $dist_t(\Sigma_2, \alpha(\Sigma_2)) \leq dist_t(\Sigma_1, t(\alpha(\Sigma_2)))$. So the first condition of $\Sigma_2 \leq_t \Sigma_1$ is proved. Thus, $\Sigma_2 \leq_t \Sigma_1$.

To show that Σ_2 is in fact strictly more elaboration-tolerant than Σ_1, assuming Σ_1 is not a tautology, it is enough to show an elaboration that is expressible in Σ_2 but is not expressible (under the translation t) in Σ_1. Let $\alpha = \{ab_\varphi \mid \varphi \in \Gamma_1\}$. Since Σ_1 is assumed not to be a tautology, there is no addition to Σ_1 that will cause it to accept all the interpretations of $\mathcal{L}_1$ (because of monotonicity). This set of interpretations[5] is yet implied by $t(\alpha(\Sigma_2))$. Thus, in this case, $\Sigma_2 <_t \Sigma_1$. ■

It is important to notice that the above theorem is not in general true for nonmonotonic theories and their equivalent monotonic counterparts. The following example demonstrates that.

Example 4.3 *Let $\Gamma_2 = \{r \vee (p \leftrightarrow q)\}$, $\mid\!\sim_C$ an entailment relation defined by $A \mid\!\sim_C \varphi \iff Circ[A; p; q] \vdash \varphi$. An equivalent monotonic theory is $\Gamma_1 = \{\neg p \wedge (q \rightarrow r)\}$. With the elaboration $\{q\}$ to Γ_1 we get $r \wedge \neg p \wedge q$. To get the equivalent theory for the nonmonotonic case, we must add both $\neg p$ and q. Adding only q will result in*

$$Circ[(r \vee (p \leftrightarrow q)) \wedge q; p; q] \equiv q \wedge (r \vee p)$$

and there is no use in trying to add either r or $\neg p$ by themselves. ■

4.3 Limits

The last two sections describe example applications of our definitions from section 2.2. We shall now examine some of the limits of our system. The following theorem intuitively says that there is no formal system that is *the most* elaboration-tolerant (Again, given a certain translation t). Notice that here we allow both axiom-adding and axiom-deleting actions.

Theorem 4.4 *Let $\mathcal{L}$ be a propositional language and t the consequence translation to $\mathcal{L}, \vdash$. Let $\Sigma = \langle \mathcal{L}, \mid\!\sim, \Gamma \rangle$ be an axiomatic formal system. Then there is another axiomatic formal system $\Sigma' = \langle \mathcal{L}, \mid\!\sim', \Gamma' \rangle$ (same language as Σ) and an elaboration $\alpha \in Actions(\mathcal{L})$ such that $\Sigma' <_{t,\alpha} \Sigma$.*

PROOF We use a combinatorial argument to show that there is Σ_{target} such that $dist(\Sigma, \Sigma_{target}) > 1$, and then find $\Sigma' \equiv_t \Sigma$ such that $dist(\Sigma', \Sigma_{target}) = 1$.

Take c to be the cardinality of the set of propositions in $\mathcal{L}$. Since Every proposition may either not show, show positively or show negatively, in every clause,

[4] We assume that the empty clause is a possible addition.

[5] Notice that the set of interpretations of $\alpha(\Sigma_2)$ is not the entire set of interpretations of $\mathcal{L}_2$.

there are 3^c possible clauses in $\mathcal{L}$ (if $\mathcal{L}$ is infinite, take the appropriate cardinals). Also, there are 2^c possible propositional models (each clause may either be true or not in each), and so there are exactly 2^{2^c} non-equivalent (equivalence measured using $\vdash$) propositional theories in $\mathcal{L}$. Thus, we have at most $2*3^c$ possible actions at our disposal (either adding or removing one of the clauses) and 2^{2^c} possible Σ_{target}'s. So, there is at least one (in fact there are many) elaboration $\alpha \in Actions(\mathcal{L})$ such that $dist_t(\Sigma, \alpha(\Sigma)) > 1$.

Take φ a clause in α that is not entailed by Σ (i.e., $\varphi \in \alpha \setminus C(\Sigma)$). Define $\mid\!\sim'$ as follows:

$$\mid\!\sim' \stackrel{def}{=} \begin{array}{l} \{\langle \Gamma, \psi \rangle \mid \psi \in \mathcal{C}(\Sigma)\} \cup \\ \{\langle \Gamma \cup \{\varphi\}, \psi \rangle \mid \psi \in \mathcal{C}(\alpha(\Sigma))\} \end{array}$$

Notice that we did not have any problem defining this entailment relation, as we did not put any restrictions on it, and most importantly, it is a *relation* between sets of axioms in $\mathcal{L}$. Now, define $\Sigma' = \langle \mathcal{L}, \mid\!\sim', \Gamma \rangle$ and sure enough, $dist_t(\Sigma', \alpha(\Sigma)) = 1$. ■

The theorem intuitively says that for every formal system Σ there is a target formal system Σ_{target} that is not easy to reach (there is an equivalent formal system to Σ that reaches Σ_{target} easier).

This proof used the fact that we put no restrictions on the entailment relation $\mid\!\sim'$, but it is not too difficult to come up with a "conservative" nonmonotonic example that will give the same proof.

5 Relation to Belief Revision

A close issue to our presentation of Elaboration Tolerance is *Belief Revision*. Belief Revision is the process that a logical theory T goes through when we wish to incorporate some new information φ (see (Alchourrón, Gärdenfors, & Makinson 1985), (Katsuno & Mendelzon 1991), (Lehmann 1995), (Antoniou 1997)). The main difference between the work done on Belief Revision and the Elaboration Tolerance treatment we presented here is that the latter is interested in the sequence of actions (e.g., syntactic additions to the knowledge base) necessary to reach a target Knowledge Base, while Belief Revision theory is interested in the change that result from adding or retracting some knowledge from the Knowledge Base.

As a result of the different motivations, there are several practical differences. Our treatment of Elaboration Tolerance is interested in *any* action for the Knowledge Base change, while Belief Revision is restricted to the actions of *add* and *remove* and their variances. Furthermore, Elaboration Tolerance is interested in the difficulty of specifying the set of actions, the length of the specification and the difficulty of executing the given change, only the last of which is interesting from the Belief Revision perspective. Another practical difference is that in Belief Revision, the underlying theory is always taken to be monotonic, and nonmonotonicity is introduced through the semantics of the revision and contraction operators. In our exposition of Elaboration Tolerance we explicitly allowed our base theory to be nonmonotonic. Last, the work on syntactic forms of Belief Revision (e.g., (Nebel 1991)) focused on Theory Base Change (Fagin, Ullman, & Vardi 1983) and the information that can be elicited from the syntactic form of the theory (mainly in the sense that axioms have a special stance among the sentences of the theory). As we saw in section 4.2, the syntactic form of a theory has an effect on its Elaboration Tolerance.

6 Conclusions

Restricting our attention to propositional languages and the actions of adding and deleting axioms to/from a knowledge base, we were able to compare systems with respect to their Elaboration Tolerance (with these allowed actions and language restrictions). For the case where only additions of axioms are allowed, we found that propositional systems with more propositional symbols are more Elaboration Tolerant, nonmonotonic systems are sometimes not more Elaboration Tolerant than an equivalent monotonic theory, there are ways to build nonmonotonic theories from monotonic ones such that the former are more Elaboration Tolerant than the latter and that there is no one most elaboration tolerant system (in our restricted scope).

Continueing the work on Elaboration Tolerance, we are currently working in three major directions. First, we wish to extend the model of knowledge-base evolution to include the fact that the knowledge engineer is not aware of the exact properties of the knowledge base when he changes it. Despite this lack of knowledge/awareness on the part of the knowledge engineer, he still manages to change the knowledge base to some degrees of success. It is our hope that this extended model will better approximate the uncertainty and difficulty of changing a knowledge base. Second, we try to make the comparison between knowledge bases more qualitative (possibly without using a measure of Elaboration Distance). The quantitative model has its virtues but it seems that we can get a different point of view when we have a qualitative model. Third, we work on finding new ways to write theories so that they are more Elaboration Tolerant. This direction benefits directly from the theorems proved in this article and other similar ones.

A generalization of the above definition to arbitrary languages and to arbitrary actions can become useful in comparing first-order systems or systems with other actions (adding preconditions to axioms, adding constant symbols and changing the entailment relation are only a few of the possibilities). These tools promise to give us new insights into the process of knowledge base construction and expansion and this work is just an initial step in that direction.

7 Acknowledgments

This work benefited greatly from discussions with Josefina Sierra, Berthe Choueiry, Alon Levy, Sasa Buvac, John McCarthy, Tom Costello, Leora Morgenstern, Ray Reiter, Fausto Giunchiglia and Shlomit Pinter. This work would not be the same without their help in various stages of the work. Leora Morgenstern, Tom Costello, Mary-Anne Williams, Fausto Giunchiglia, Alon Levy, Shlomit Pinter and Erika Henik read drafts of this work, and I am grateful for their remarks. Last, but not least, Josefina Sierra read many versions, listened to many complaints and difficulties and commented during various stages of the ongoing work. Without her support I am doubtful this work would have been in the shape it is now.

This research was supported by an ARPA (ONR) grant N00014-94-1-0775 and an Air Force grant F49620-97-1-0207.

References

Alchourrón, C. E.; Gärdenfors, P.; and Makinson, D. 1985. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic* 50:510–530.

Amir, E. 1997. Machinery for elaborating action – Preliminary Report. In NRAC-97' and in http://www-formal.stanford.edu/eyal/nmr/machinery-ea.ps.

Antoniou, G. 1997. *Nonmonotonic Reasoning*. Cambridge, Massachusetts: MIT Press.

Brachman, R., and Levesque, H. 1985. *Readings in Knowledge Representation*. Morgan Kaufmann.

Costello, T. 1997. Beyond minimizing change. In *Proceedings of AAAI-97*.

Fagin, R.; Ullman, J. D.; and Vardi, M. Y. 1983. On the semantics of updates in databases. In *Proceedings of the Second ACM SIGACT-SIGMOD Symposium on Principles of Database Systems*, 352–365.

Giunchiglia, F., and Walsh, T. 1992. A theory of abstraction. *Artificial Intelligence* 57(2-3):323–389.

Katsuno, H., and Mendelzon, A. O. 1991. Propositional knowledge base revision and minimal change. *Artificial Intelligence* 52(KRR-TR-90-3):263–294.

Lehmann, D. 1995. Belief revision, revised. In *IJCAI-95*, 1534–1540.

McCarthy, J. 1980. Circumscription—A Form of Non-Monotonic Reasoning. *Artificial Intelligence* 13:27–39. Reprinted in (McCarthy 1990).

McCarthy, J. 1988. Mathematical logic in artificial intelligence. *Daedalus* 117(1):297–311.

McCarthy, J. 1990. *Formalization of common sense, papers by John McCarthy edited by V. Lifschitz*. Ablex.

McCarthy, J. 1998. Elaboration Tolerance. In CommonSense '98 and in McCarthy's web page http://www-formal.stanford.edu/jmc/elaboration.html.

Minsky, M. 1975. A framework for representing knowledge. Winston.

Moore, R. C. 1987. Possible-world semantics for autoepistemic logic. In Ginsberg, M., ed., *Readings in nonmonotonic reasoning*. San Mateo, CA: Morgan Kaufmann. 137–142.

Nebel, B. 1991. Belief revision and default reasoning: syntax-based approaches. In *kr91*. SF: MK. 417–428.

Reiter, R. 1980. A Logic for Default Reasoning. *Artificial Intelligence* 13 (1–2):81–132.

Shanahan, M. P. 1996. Robotics and the common sense informatic situation. In *Proceedings ECAI 96*, 684–688.

Shoenfield, J. R. 1967. *Mathematical Logic*. Duke University: Addison-Wesley.