

Modular First-Order Ontologies via Repositories

Michael GRÜNINGER ^a, Torsten HAHMANN ^b, Ali HASHEMI ^a, Darren ONG ^a,
Atalay OZGOVDE ^b

^a*Department of Mechanical and Industrial Engineering, University of Toronto,
Toronto, Ontario, Canada M5S 3G8*

^b*Department of Computer Science, University of Toronto, Toronto, Ontario, Canada
M5S 3G8*

Abstract.

From its inception, the focus of ontological engineering has been to support the reusability and shareability of ontologies, as well as interoperability of ontology-based software systems. Among the approaches employed to address these challenges have been ontology repositories and the modularization of ontologies. In this paper we combine these approaches and use the relationships between first-order ontologies within a repository (such as non-conservative extension and relative interpretation) to characterize the criteria for modularity. In particular, we introduce the notion of core hierarchies, which are sets of theories with the same non-logical lexicons and which are all non-conservative extensions of a unique root theory. The technique of relative interpretation leads to the notion of reducibility of a theory to a set of theories in different core hierarchies. We show how these relationships support a semi-automated procedure that decomposes an ontology into irreducible modules. We also propose a semi-automated procedure that can use the relationships between modules to characterize which modules can be shared and reused among different ontologies.

Keywords. ontology repository, modularity, first-order logic, non-conservative extension, conservative extension, reducibility, similarity

1. Introduction

Ontology repositories have been proposed as part of the infrastructure required to support interoperability of ontology-based software systems through the reusability and shareability of ontologies. We are naturally led to the question as to the modular nature of the constituent ontologies within such a repository. If a repository consists of a set of ontologies, can the repository provide a way of modularizing its constituent ontologies? What criteria do we use to identify which modules are atomic, so that they form the smallest building blocks for assembling new modular ontologies?

For the purpose of this paper, we consider ontologies as axiomatizations of some concept(s) in the language of first-order logic; each ontology gives rise to a theory as the closure of the ontology's axiomatization under logical consequences (entailments). A module of an ontology is then an axiomatization of a subtheory of the ontology. One can take an "internal" view of modules in the sense that an ontology is explicitly con-

structed by extending and combining existing ontologies in the repository. An alternative “external” view decomposes an ontology based on logical relationships of its subtheories to existing modules in the repository. In this paper we explore what logical relationships among a set of first-order ontologies can be used as basis for the decomposition of ontologies following this external view and how these relationships can be used. We identify two levels of modularity that arise from these logical relationships.

One fundamental logical relationship is non-conservative extension; this leads in Section 2 to the notion of a hierarchy of ontologies that have the same non-logical lexicon. The resulting first approach to modularity decomposes an ontology into atomic modules in a single hierarchy based on similarities and differences between the ontology and existing atomic modules in that hierarchy. Each ontology in a hierarchy is then logically equivalent to a set of atomic modules.

For ontologies that are axiomatized using disjoint non-logical lexicons¹, we need to determine whether or not the non-logical lexicon of one ontology can be interpreted in the non-logical lexicon of other ontologies in the repository. This leads in Section 3 to our second approach to modularity, based on the relationship of reducibility, in which one ontology is definably equivalent to the union of existing modules in different hierarchies.

We define a repository to be a set of hierarchies that are ordered by the reducibility relationship; the hierarchies whose constituent modules are not all reducible to other modules in the repository are the core hierarchies and they form the modular building blocks of the ontologies in the repository. Each core hierarchy has a common root theory that captures inherent properties of the axiomatized concept. E.g. all ontologies in a repository that axiomatize the notion of “time” are some kind of linear order. Each atomic module non-conservatively extends the root theory but no other theory in the hierarchy, and specifies an ontological design choice, which captures a property of all its models not commonly accepted by all ontologies within the same hierarchy. E.g. there are time ontologies that are dense and others that are discrete (and as such non-dense), so density is an ontological design choice for time ontologies. In this pragmatic approach not all theoretically feasible ontological design choices are extracted, but only the ones that are used by at least one of the ontologies in the hierarchy. We can then view each ontology contained in a single core hierarchy of the repository as an axiomatization of the root theory of the hierarchy extended by a set of additional ontological design choices. Thereby, we give a more formal perspective on the informal notion of design choices.

We consider repositories to be dynamic ontology design artefacts, that is, the modularization of the constituent ontologies evolves with the addition of more ontologies. We propose semi-automated procedures for the decomposition and refinement of ontologies that rely on the existing modules in the repository, giving practical guidance for modularizing ontologies and designing new modular ontologies. The set of procedures in Section 6 identify modules of an ontology that are all equivalent to modules in core hierarchies which provides a modularization of the ontology. The set of procedures in Section 7 update a single hierarchy with a new ontology that has the same non-logical lexicon. The new ontology is decomposed into a set of independent ontological design choices based on the other modules in that hierarchy. Through these procedures previously atomic ontological design choices may be split into a set of more fine-grained design choices and completely new design choices may be introduced. These procedures

¹If ontologies do not use exactly the same non-logical lexicons but share some non-lexical symbols, those shared can be easily renamed to ensure that the ontologies use strictly disjoint non-logical lexicons.

relate closely to the two kinds of modularity. The first set of procedures divide a new ontology into modules in different hierarchies while the second set of procedures divide those modules into atomic modules within each hierarchy.

2. Hierarchies

Before we define what it means for a set of ontologies (or their resulting theories) to be in a hierarchy, we introduce some preliminary notions that are used throughout the paper. These are well-known definitions in mathematical logic².

Definition 1 A first-order theory is a set of first-order sentences closed under logical entailment.

Definition 2 Let T be a first-order theory.

The non-logical lexicon (signature) of T , denoted by $\Sigma(T)$, is the set of all constant symbols, function symbols, and relation symbols that are used in T .

The language of T , denoted by $\mathcal{L}(T)$, is the set of all first-order formulas that only use the non-logical symbols in the signature $\Sigma(T)$.

Definition 3 Let T_1, T_2 be two first-order theories such that $\Sigma(T_1) \subseteq \Sigma(T_2)$.

We say that T_2 is an extension of T_1 iff for any sentence $\sigma \in \mathcal{L}(T_1)$,

$$\text{if } T_1 \models \sigma \text{ then } T_2 \models \sigma.$$

T_2 is a conservative extension of T_1 iff for any sentence $\sigma \in \mathcal{L}(T_1)$,

$$T_2 \models \sigma \text{ iff } T_1 \models \sigma.$$

T_2 is a non-conservative extension of T_1 iff T_2 is an extension of T_1 and there exists a sentence $\sigma \in \Sigma(T_1)$ so that

$$T_1 \not\models \sigma \text{ and } T_2 \models \sigma.$$

A first-order ontology is a set of first-order sentences (axioms) that characterize a first-order theory, which is the closure of the ontology's axioms under logical entailment. In the rest of the paper we will simply drop the term first-order and assume ontologies and theories to be first-order.

Two ontologies O_1 and O_2 that use the same non-logical lexicon Σ have logically equivalent theories if for all sentences σ expressed in Σ

$$O_1 \models \sigma \iff O_2 \models \sigma$$

We do not distinguish ontologies with logically equivalent theories. Even though ontologies stored in an actual repository are concrete sets of axioms, we will discuss our ideas in this paper in terms of the more abstract *repository of theories*. In other words,

²For additional background for the notions of structures and models, we refer the reader to [17].

the concepts described in this paper do not depend on how the theories are axiomatized. Moreover, we do not distinguish logically equivalent theories with the same non-logical lexicon. Thus, for the purposes of this paper we assume that each ontology can be uniquely mapped to a theory and vice versa and that two logically equivalent theories are identified by a unique ontology.

We also do not attempt to formally define the notion of an ontology. Although every module is a set of axioms, not every module may be seen as an ontology from the point of view of the modeller. Semantically, a module can simply be a reusable building block for an ontology without capturing a meaningful metaphysical notion in its own right.

Now, the first step is to define sets of theories of the same concept which can be characterized by combinations of ontological design choices. Two theories can then be distinguished by their different sets of design choices. For example, the theory T_{lp_dense} ³ consists of the two atomic design choices of the order being linear dense. Each such choice may manifest itself in multiple alternative sets of axioms; there may not be a one-to-one mapping between design choices and concrete axioms of an ontology. While each design choice may be written as an axiom – a so-called logical choice, not every logical choice is an ontological design choice in its own right. The objective is to support the comparability of theories within a hierarchy with respect to their ontological choices, and in this way to explicitly identify their similarities and differences (the axioms in which they differ from all the theories they extend). In this section we will formally define the notions of hierarchy, root theory, trunk theory, similarity, and difference.

2.1. The Role of Non-conservative Extensions

If theories are characterized by sets of design choices, they can be ordered by the presence or absence of those design choices. In order for the design choices to be comparable, their axiomatizations need to be expressed using the same signature. Using these intuitions, we can define an ordering over a set of theories:

Definition 4 A *hierarchy* $\mathbb{H} = \langle \mathcal{H}, \leq \rangle$ is a partially ordered, finite set of theories $\mathcal{H} = T_1, \dots, T_n$ such that

1. $\Sigma(T_i) = \Sigma(T_j)$, for all i, j ;
2. $T_1 \leq T_2$ iff T_2 is an extension of T_1 ;
3. $T_1 < T_2$ iff T_2 is a non-conservative extension of T_1 .

Hierarchies are obviously not closed under subtheories since infinitely many subtheories exist but we can define the following notion:

Definition 5 Let $\mathbb{H} = \langle \mathcal{H}, \leq \rangle$ be a hierarchy. We call a theory T *compatible* with the hierarchy \mathbb{H} iff $\Sigma(T) = \Sigma(T_i)$ for any $T_i \in \mathcal{H}$.

Compatible theories may not be in the hierarchy, but all theories in a hierarchy are also compatible with the hierarchy. We must also be aware that for a given theory T in

³We adopt the following naming convention for the online CLIF axiomatization of theories – each theory is associated with a URI such that the subscript in the theory name is the unqualified name in the URI. For example, the CLIF axiomatization of the theory T_{lp_dense} can be found at <http://stl.mie.utoronto.ca/colore/time/lp-dense.clif>

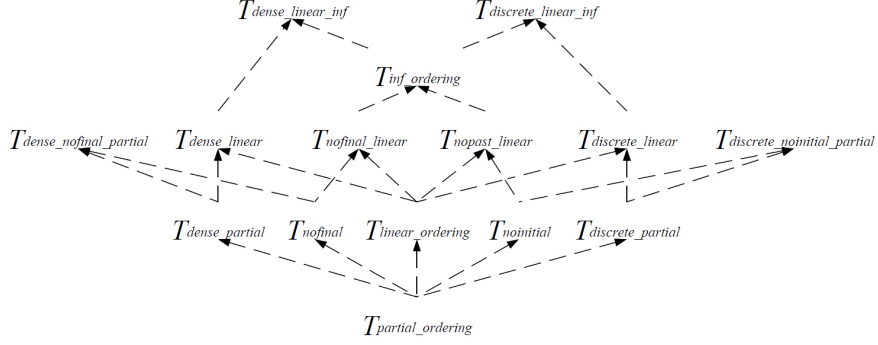


Figure 1. Hierarchy of Ordering theories. Dashed lines denote non-conservative extensions.

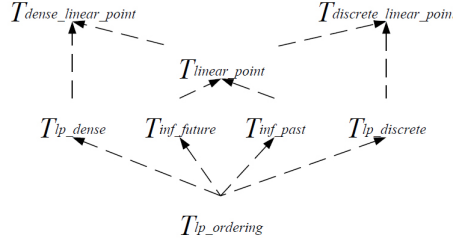


Figure 2. Hierarchy of Timepoint theories. Dashed lines denote non-conservative extensions.

a hierarchy, a subtheory S of T in the hierarchy may not be axiomatizable by a subset of the axioms of T . In other words, the partial order over theories within an hierarchy cannot be established solely by relying on the subset relations between sets of axioms.

Consider the hierarchy of ordering theories shown in Figure 1, with theories ordered such that ones higher in the hierarchy non-conservatively extend those they are connected to below. For example, $T_{linear_ordering}$ is the theory of linearly ordered points and T_{dense_linear} is the theory of dense linearly ordered points, so following Definition 1 we have $T_{dense_linear} > T_{linear_ordering}$ since every sentence entailed by $T_{linear_ordering}$ is also entailed by T_{dense_linear} and there exist sentences entailed by T_{dense_linear} that are not entailed by $T_{linear_ordering}$. Similarly, for the hierarchy of timepoint theories shown in Figure 2, we have $T_{dense_linear_point} > T_{lp_dense}$ since every sentence entailed by T_{lp_dense} , the theory of dense linearly ordered timepoints, is also entailed by $T_{dense_linear_point}$, the theory of dense linearly ordered timepoints that extends infinitely into the past and future, but not vice versa.

Although the ordering over theories in a hierarchy is defined with respect to entailment, the two notions are distinct. For example, consider the standard theory $Th(\mathcal{N}, 0, S)$ of natural numbers vs. the theory $Th(\mathcal{N}, 0, S, lt)$ of natural numbers with an additional ordering relation lt . We have

$$Th(\mathcal{N}, 0, S, lt) \models Th(\mathcal{N}, 0, S)$$

yet we do not have $Th(\mathcal{N}, 0, S, lt) \leq Th(\mathcal{N}, 0, S)$, since these two theories do not share a signature and hence cannot be in the same hierarchy.

Let $Mod(T)$ denote the set of models of a theory T . Given two theories T_1 and T_2 such that $T_2 \models T_1$, we do not in general have $Mod(T_2) \subseteq Mod(T_1)$, since the models of T_2 may not be models of T_1 . For example, suppose that T_1 is the theory of bipartite incidence structures and T_2 is the theory of ordered geometries. Models of T_2 are not isomorphic to models of T_1 ; rather, they contain substructures that are isomorphic to models of T_1 . On the other hand, if we restrict ourselves to theories within the same hierarchy, then we can use subsets of models to characterize the relationships between theories:

Lemma 1 *If T_1 and T_2 are theories in the same hierarchy, then*

$$T_1 < T_2 \Leftrightarrow Mod(T_2) \subsetneq Mod(T_1).$$

Proof: \Rightarrow : Assume that $T_1 < T_2$. Now suppose there exists a model $\mathcal{M} \in Mod(T_2)$ that is not a model of T_1 , i.e. $\mathcal{M} \notin Mod(T_1)$. There exists a sentence σ so that σ is satisfied by \mathcal{M} and σ is not consistent with T_1 , i.e. $T_1 \models \neg\sigma$. Since $T_2 \not\models \neg\sigma$, $T_1 \not\leq T_2$, a contradiction.

Moreover, $T_2 \not\leq T_1$, i.e. there exists a sentence σ such that $T_2 \models \sigma$ but $T_1 \not\models \sigma$. There exists a model of T_1 that satisfies $\neg\sigma$ but which cannot be a model of T_2 . Hence $Mod(T_2) \subsetneq Mod(T_1)$.

\Leftarrow : If $T_1 \not\leq T_2$, then there exists a sentence σ such that $T_1 \models \sigma$ but $T_2 \not\models \sigma$. A model of T_2 exists that satisfies $\neg\sigma$ but cannot be a model of T_1 . This contradicts the assumption that $Mod(T_2) \subsetneq Mod(T_1)$.

Moreover, there must be a model of T_1 that is not a model of T_2 . Let σ capture this model which is consistent with T_1 but not consistent with T_2 . Thus, $T_2 \models \neg\sigma$ while $T_1 \not\models \neg\sigma$, hence $T_2 \not\leq T_1$. \square

In other words, models of a theory in a hierarchy are models of its subtheories that are in the same hierarchy; extensions only restrict the sets of models, not the structures of the models themselves. This is not in general the case for all theories; for example, the theories in the PSL Ontology [12] do not form a hierarchy, since each theory expands the non-logical lexicon; as a result, the models of a theory such as the PSL Ontology are constructed by combining and extending models of other theories.

Root theories are the theories that do not extend any other theory in the hierarchy.

Definition 6 *A theory T in a hierarchy is a root theory iff it does not non-conservatively extend any other theory in the same hierarchy.*

In the ordering hierarchy, $T_{partial_ordering}$ is the only root theory as it does not extend any other theory in its hierarchy and for the timepoints hierarchy the only root theory is $T_{lp_ordering}$ for the same reason. Though in general multiple root theories may exist, we later define a more constrained type of hierarchy – a closed hierarchy – that requires the root theory to be uniquely determined up to logical equivalence.

For two theories T_1, T_2 with the same signature Σ we say that T_2 is stronger than T_1 (or, equally, that T_1 is weaker than T_2) if T_2 is a non-conservative extension of T_1 independent of whether T_1 and T_2 are actually contained in a hierarchy. Obviously, if $T_2 > T_1$ for theories T_1, T_2 in the same hierarchy, then T_2 is stronger than T_1 .

2.2. Similarity between Theories in a Hierarchy

Within a hierarchy, all theories are partially ordered by their sets of theorems. In the following we want to extract a specific subset S of the theorems shared by two theories T_1 and T_2 within a hierarchy. We refer to this subset of shared inferred sentences, which itself is a theory with signature $\Sigma(T_1)$, as the similarity of T_1 and T_2 , while theorems that T_1 and T_2 differ in are captured as remainders (with respect to the similarity S). Again, this notion of similarity is defined with respect to subtheories and not with respect to sets of axioms; however, it still varies from the idea discussed in [15].

Obviously, two theories in the same hierarchy have a common strongest subtheory (up to logical equivalence). However, this subtheory may be overly specific. Consider for example the theories $T_{nofinal}$ and $T_{noinitial}$ in the hierarchy of ordering theories in Fig. 1:

$$\begin{aligned} T_{nofinal} &\models \forall x \exists y (x < y) \\ T_{noinitial} &\models \forall x \exists y (y < x) \end{aligned}$$

Either sentence corresponds to some intuitive design choice for orderings: whether there is some initial (smallest) element and whether there is some final (largest) element. This is maybe better illustrated by the same entailments in the two theories of timepoints T_{inf_future} and T_{inf_past} . In some applications (such as planning) it may be suitable to assume that for every timepoint in the future there exists another timepoint even further in the future. In an application that only reasons about the past (such as dating of ancient artefacts) such assumption may not be useful, but instead we may want to assume that there is a timeline that reaches infinitely into the past. Those two assumptions are independent design choices. Thus their shared theory should make no claim at all about whether a smallest (or earliest) or greatest (latest) element exists. In particular, the similarity of $T_{nofinal}$ and $T_{noinitial}$ (and likewise of T_{inf_future} and T_{inf_past}) should not entail the disjunctive sentence

$$[\forall x \exists y (x < y)] \vee [\forall x \exists y (y < x)] \quad (*)$$

even though it is entailed by both $T_{nofinal}$ and $T_{noinitial}$. The reason is that $T_{nofinal}$ and $T_{noinitial}$ already entail disjoint stronger sentences which better capture their respective ontological choices. Hence, our definition of similarity is not simply the strongest subtheory of two given theories in the same hierarchy:

Definition 7 Let T_1 and T_2 be theories in the same hierarchy with the signature Σ .

The similarity between T_1 and T_2 is the strongest theory (up to logical equivalence) $S \subseteq T_1 \cap T_2$ with $\Sigma(S) = \Sigma(T_1)$ so that for all $\sigma, \omega \in \mathcal{L}(T_1)$ if

$$T_1 \models \sigma \quad \text{and} \quad T_2 \models \omega \quad \text{and} \quad S \not\models \sigma \quad \text{and} \quad S \not\models \omega$$

then either $\sigma \vee \omega$ is independent of S or $\sigma \vee \omega$ is a tautology.

In other words, S contains all sentences shared by T_1 and T_2 except disjunctive sentences $\sigma \vee \omega$ so that T_1 entails exactly one of σ and ω while T_2 entails only the

other. In this sense, we rid the similarity from any unnecessarily specific new disjunctive sentences. In our previous example, choose $T_1 = T_{nofinal}$, $T_2 = T_{noinitial}$, $\sigma = \forall x \exists y (x < y)$, and $\omega = \forall x \exists y (y < x)$. Since $T_1 \models \sigma$, $T_2 \not\models \sigma$, $T_2 \models \omega$, and $T_1 \not\models \omega$, the sentence (*) cannot be entailed by the similarity S since it is not a tautology and thus must be independent of S . Intuitively, the similarity of $T_{nofinal}$ and $T_{noinitial}$ is the common subtheory that also allows models without greatest element and without initial element.

We now confirm that a similarity always exists for two theories in the same hierarchy. Then, by the Definition 7 the similarity must be unique.

Lemma 2 *The similarity between two theories T_1, T_2 in the same hierarchy exists.*

Proof: Suppose there does not exist a similarity between T_1 and T_2 . This can be the case only in one of the two following cases. (1) If T_1 and T_2 have no common subtheory, or (2) if T_1 and T_2 have two subtheories S and S' satisfying the last condition of Def. 7 but neither one is stronger than the other.

- (1) If no stronger theory satisfies the definition of similarity, then the empty theory⁴, which is unique, is the strongest subtheory of T_1 and T_2 .
- (2) We will derive a contradiction for this case. It requires that two sentences $\sigma, \omega \in \mathcal{L}(T_1)$ exists such that $S \models \sigma$, $S' \not\models \sigma$, $S' \models \omega$, $S \not\models \omega$, and

$$S \equiv (S \cap S') \cup \sigma$$

and

$$S' \equiv (S \cap S') \cup \omega.$$

Because $S \models \sigma$ and $S \subseteq T_1 \cap T_2$ the sentence σ and any disjunction of the form $\sigma \vee \tau$, where τ is an arbitrary sentence in the language $\mathcal{L}(T_1)$, do not need to be independent of $S' \subseteq T_1 \cap T_2$ and, thus, $\sigma \in S'$ by Def. 7. This contradicts our initial assumption that $S' \not\models \sigma$. Likewise, we can argue that $\omega \in S$, which contradicts $S \not\models \omega$. Therefore we must have

$$S \equiv (S \cap S') \cup \sigma \cup \omega \equiv (S \cap S') \cup \omega \cup \sigma \equiv S'.$$

Hence, a strongest theory as defined by Def. 7 is unique and thus the similarity always exists for two theories in the same hierarchy. \square

Note that for general hierarchies, the similarity of two theories in the same hierarchy is not necessarily contained in the hierarchy. Only in closed hierarchies as defined in Section 2.3, the similarity must be itself a theory of the hierarchy.

Note further that two inconsistent theories in the same hierarchy do not necessarily have the empty theory as similarity. For example, the theory T_{dense_linear} of dense linearly ordered points and the theory $T_{discrete_linear}$ of discrete linearly ordered points are inconsistent. While T_{dense_linear} entails density between points, $T_{discrete_linear}$ entails discreteness between points. Their similarity is the strongest subtheory of both

⁴The ‘empty theory’ contains only first-order validities, but no other sentences.

T_{dense_linear} and $T_{discrete_linear}$ that does neither entail density nor discreteness. This theory, $T_{linear_ordering}$, is one of linearly ordered points independent of the notions of density or discreteness following Def. 7. Note that the similarity cannot always be constructed as the intersection of the axioms of two ontologies.

2.3. Closed Atomistic Hierarchies

We can now use the notion of similarity to define when a hierarchy is closed and show that a closed hierarchy has a unique root theory.

Definition 8 A hierarchy $\mathbb{H} = \langle \mathcal{H}, \leq \rangle$ is closed iff the similarity S between any two theories $T_1, T_2 \in \mathcal{H}$ is non-empty and $S \in \mathcal{H}$.

Since a hierarchy is a partially ordered set, this basically states that a closed hierarchy is a meet-semilattice, that is, the meet (similarity) of any two theories in the hierarchy is itself in the hierarchy. The following verifies that a closed hierarchy has a unique smallest element, the so-called root theory.

Lemma 3 A closed hierarchy has a unique root theory up to logical equivalence.

Proof: Suppose two non-logically equivalent roots T_1 and T_2 exist in a hierarchy \mathcal{H} . Since the hierarchy is closed, the similarity S between T_1 and T_2 is non-empty and in \mathcal{H} . Both T_1 and T_2 extend S , thus S is the new root theory while T_1 and T_2 cannot be root theories in the hierarchy. \square

Intuitively, the root theory of a hierarchy axiomatizes an elementary class of structures; each theory in the hierarchy axiomatizes a subclass of this class of structures. For example, the theories that axiomatize classes of (non-strict) partial orderings form a hierarchy – all theories have the same signature (i.e. the ordering relation \leq) and all are extensions of the theory that contains the three axioms for a non-strict partial ordering (transitivity, reflexivity, and antisymmetry).

Definition 9 Let $\langle \mathcal{H}, \leq \rangle$ be a hierarchy with $T_1, T_2 \in \mathcal{H}$. If $T_1 < T_2$ and no other theory $T \in \mathcal{H}$ exists so that $T_1 < T < T_2$, we say T_2 covers T_1 and write $T_1 \prec T_2$.

Since a closed hierarchy contains a unique root theory, the class of theories that cover the root theory will play a key role.

Definition 10 A trunk theory T is a theory that covers the unique root theory of a closed hierarchy $\langle \mathcal{H}, \leq \rangle$.

The trunk theories are the atoms in the partially ordered set that the hierarchy forms. By the way we defined similarity, the different trunk theories of a closed hierarchy each axiomatize a single (atomic) ontological design choice that extends the root theory. For example, the theory of dense partial orderings and the theory of discrete partial orderings form trunk theories for the hierarchy associated with partial orderings. The former introduces the ontological design choice “density” while the latter introduced the ontological design choice of “discreteness”. This allows a more formal definition of the informal concept of ontological design choices: Any remainder between a trunk theory and the

root theory in a hierarchy of the repository axiomatizes the atomic design choice made by a particular trunk theory.

We can now verify that the similarity between trunk theories in the same closed hierarchy is the unique root theory.

Lemma 4 *The similarity between any two trunk theories of the same closed hierarchy is logically equivalent to the root theory.*

We can introduce an additional property for hierarchies which enforces the condition that every theory in the hierarchy makes a set of atomic design choices. The hierarchy is thereby required to be atomistic, i.e. each theory is the join of a set of atoms (trunk theories).

Definition 11 *A closed hierarchy $\mathbb{H} = \langle \mathcal{H}, \leq \rangle$ is atomistic iff every $T \in \mathcal{H}$ that is not a trunk theory satisfies the following property: if $T_1, \dots, T_n \in \mathcal{H}$ are all the theories covered by T , then T is logically equivalent to $T_1 \cup \dots \cup T_n$.*

Note that this allows for multiple logically equivalent axiomatizations, that is, the set of theories covered by T is not necessarily the minimal set of theories whose union is logically equivalent to T . We can define a minimal set by only including trunk theories:

Definition 12 *A set of trunk theories T_1, \dots, T_n is a profile of a theory T iff T is logically equivalent to $T_1 \cup \dots \cup T_n$.*

We obtain the following theorem similar to what is known about atomistic lattices:

Theorem 1 *A non-root theory T in a closed atomistic hierarchy has a unique profile up to logical equivalence.*

Proof: Proof by induction over the distance in the partial order $<$ from a trunk theory within the hierarchy $\mathbb{H} = \langle \mathcal{H}, \leq \rangle$.

Base case: Each trunk theory $T \in \mathcal{H}$ has itself as its unique profile, since no other trunk theory is entailed by or entails T (Lemma 4).

Induction step: Assume that T_1, \dots, T_n have unique profiles, and T covers a subset thereof, that is, T covers all of T_1, \dots, T_k with $k \leq n$. Then T is by the definition of an atomistic hierarchy logically equivalent to the union of the profiles of T_1, \dots, T_k , which in turn by the definition of profile is the unique profile of T .

Consequently every theory in \mathcal{H} has a unique profile. \square

For example, the ordering hierarchy is atomistic. E.g., the set of three trunk theories $T_{nofinal}$, $T_{noinitial}$, and $T_{linear_ordering}$, make up the profile of $T_{inf_ordering}$. Since $T_{nofinal}$ is the theory of partial orders without a final point, $T_{noinitial}$ is the theory of partial orders without an initial point, and $T_{linear_ordering}$ is the theory of linear orders, it is easy to see that their union is logically equivalent to the theory of infinite linear orders, $T_{linear_ordering}$, and thus the three theories indeed form the profile of $T_{linear_ordering}$. This example illustrates how the profile modularizes a larger theory in the hierarchy into a set of atomic theories.

Theorem 2 Let $\mathbb{H} = \langle \mathcal{H}, \leq \rangle$ be a closed atomistic hierarchy.

The similarity S of two theories $T_1, T_2 \in \mathcal{H}$ is logically equivalent to the union of the trunk theories that are the intersection of the profile of T_1 and the profile of T_2 .

Proof: Since the hierarchy is closed, the similarity S of T_1 and T_2 is in the hierarchy.

Moreover, the hierarchy is atomistic and thus by Thm. 2 S has a unique profile in \mathcal{H} . Then it is sufficient to prove that the profile of the similarity is equivalent to the intersection of the profiles of T_1 and T_2 . We show that $profile(T_1) \cap profile(T_2) \supseteq profile(S)$ and $profile(T_1) \cap profile(T_2) \subseteq profile(S)$.

$profile(T_1) \cap profile(T_2) \supseteq profile(S)$: Suppose some trunk theory T is in the profile of S , then both $T_1 \models T$ and $T_2 \models T$. Since by Lemma 4 the similarity of any two trunk theories is the root theory, no other trunk theory can entail T . But for T_1 and T_2 to be equivalent to their respective profiles, each of the profiles of T_1 and T_2 must entail T . Hence T is in the profiles of both T_1 and T_2 , and thus in the intersection of their profiles.

$profile(T_1) \cap profile(T_2) \subseteq profile(S)$: Since the hierarchy is closed, the similarity S between T_1 and T_2 is also in \mathcal{H} . Since the hierarchy is atomistic, by Thm. 1 S has a unique profile in \mathcal{H} . Now it is easy to see that any trunk theory that is both in the profiles of T_1 and T_2 , must also be in the profile of S . \square

In the previous example, the profile of T_{dense_linear} is the set of trunk theories of $T_{dense_partial}$ and of $T_{linear_ordering}$, and the profile of $T_{discrete_linear}$ is the set of trunk theories $T_{discrete_partial}$ and $T_{linear_ordering}$. We recall that the similarity of these two theories is a theory of linearly ordered points that makes no commitment with respect to density or discreteness, $T_{linear_ordering}$, which is equivalent to the intersection of their profiles.

2.4. Hierarchies and Interoperability

Ontologies often play a key role in ensuring interoperability between software systems. We can talk about the interoperability of two software systems in terms of the relationships of their underlying ontologies. We say that two software systems can be semantically integrated if the sets of intended models of their (explicit or implicit) ontologies are equivalent. However, systems cannot exchange the models of their ontologies themselves – they can only exchange sentences in the formal language (with a given signature) that they use to represent their knowledge. Suppose that the software systems use as their underlying ontologies two different theories within the same closed hierarchy. How can we use the relationships among the theories in the repository to articulate the nature of interoperability between the software systems relying on those theories?

If the theories of two software systems are logically equivalent, then obviously the two software systems can exchange arbitrary sentences and preserve theorems. If the two theories are not logically equivalent, then their similarity is the shared subtheory that characterizes a set of theorems that can be shared. But how can we capture the sentences that are not in the similarity but are entailed by one of the theories, that is, the sentences that are not shared? If these sentences are mutually inconsistent, then they capture the obstacles to semantic interoperability between the software system. On the other hand, if these sentences that are different among the two theories are mutually consistent, then it is possible for one or the other system to consider extending its theory in order to share

a larger subtheory. In the remainder of this section, we will use the notion of similarity among theories in a hierarchy to explore a variety of metatheoretic relations between theories through which we can compare theories in such a way as to support shareability and reusability.

2.4.1. Remainders

The first metatheoretic relation we consider is between a theory T_1 and one of its non-conservative extensions T_2 .

Definition 13 Let T_1, T_2 be theories in the same hierarchy such that $T_1 < T_2$.

The *remainder* between T_2 and T_1 is the weakest theory (up to logical equivalence) T' with $\Sigma(T') = \Sigma(T_1)$ so that

$$T_2 \equiv T_1 \cup T'.$$

Intuitively, the remainder constitutes the weakest axiomatization that suffices to extend T_1 to T_2 ; in particular it must contain the ontological design choices of T_2 that are not contained in T_1 . As such, this set of axioms may not be itself in the hierarchy or the entire repository. In fact, the next lemma shows that remainders do not extend the common root theory in a closed hierarchy and are therefore not themselves in the hierarchy if the hierarchy is closed.

Lemma 5 Let T_1, T_2 be theories in the same closed hierarchy with $T_1 < T_2$. The remainder between T_2 and T_1 is non-empty and is not in the closed hierarchy.

Proof: Since $T_1 < T_2$, that is, $Mod(T_2) \subsetneq Mod(T_1)$, a sentence σ capturing a model in $Mod(T_1) \setminus Mod(T_2)$ is not consistent with T_2 , that is $T_2 \models \neg\sigma$, but is consistent with T_1 , that is, $T_1 \not\models \neg\sigma$. Then by the definition of remainder, $T' \models \neg\sigma$, therefore T' is non-empty.

The closed hierarchy has a unique root T^{root} by Lemma 3. Suppose T' is in the hierarchy, then it must extend T^{root} . But then, by the definition of the remainder, there exists a theory T'' that is weaker than T' which does not entail T^{root} . Then T' is not the remainder. \square

Consider the theories of linearly ordered points, $T_{linear_ordering}$, and of partially ordered points, $T_{partial_ordering}$. The minimal set of axioms required to extend $T_{partial_ordering}$ to $T_{linear_ordering}$ is the axiom expressing that the order is total and as such, that axiom constitutes the remainder. However, that single axiom alone does not extend the root theory and therefore is not in the closed hierarchy of ordering theories.

2.4.2. Differences

Though remainders are not included in closed hierarchies, they lend themselves to a definition of differences between a theory and a non-conservative extension thereof.

Definition 14 Let T_1, T_2 be theories in the same closed hierarchy \mathbb{H} so that $T_1 < T_2$. Let T' be the remainder between T_2 and T_1 and let T^{root} be the unique root theory of \mathbb{H} .

The difference between T_2 and T_1 is the theory (up to logical equivalence) T^d with $\Sigma(T^d) = \overline{\Sigma(T_1)}$ so that

$$T^d \equiv T^{root} \cup T'.$$

For example, the difference of T_{dense_linear} and $T_{linear_ordering}$ in the ordering hierarchy is the theory that extends the root theory with the remainder between T_{dense_linear} and $T_{linear_ordering}$. The remainder between those two theories is the axiom that enforces density between points; extending the root theory $T_{partial_ordering}$ with this remainder yields the difference $T_{dense_partial}$.

It is easy to see that T^d non-conservatively extends T^{root} and that T_2 non-conservatively extends T^d . Moreover, non-empty differences always exist by Def. 14.

Even though differences may not be included in a hierarchy, they are always compatible with the hierarchy. In closed atomistic hierarchies we can easily obtain the difference between any two constituent theories from the trunk theories of the hierarchy as follows.

Theorem 3 Let T_1, T_2 be theories in the same closed atomistic hierarchy with $T_1 < T_2$.

The difference T^d between T_2 and T_1 is logically equivalent to the set difference of the profile of T_2 and the profile of T_1 .

Proof: Let the trunk theories T_{21}, \dots, T_{2n} be the profile of T_2 , that is $T_2 \equiv T_{21} \cup \dots \cup T_{2n}$.

The profiles of T_1 and T^d are then subsets of the profile of T_2 . Suppose some T_{2i} is in the profiles of T_1 and T^d . Then T_{2i} is in the similarity of T_1 and T^d and by Lemma 3, $T^{root} \models T_{2i}$, so T_{2i} is not a trunk theory – a contradiction. Thus the profiles are disjoint subsets of $profile(T_2)$.

Now suppose there is a trunk theory T_{2i} in the profile of T_2 that is neither in the profile of T_1 nor of T^d , that is $T_1 \not\models T_{2i}$ and $T^d \not\models T_{2i}$. With $T_2 \models T_{2i}$, this violates the definition of difference (with the definition of remainder). \square

For example, the profile of T_{dense_linear} is the set of trunk theories $T_{dense_partial}$ and $T_{linear_ordering}$, and the profile of trunk theory $T_{linear_ordering}$ is itself. As described in an earlier example, the difference between T_{dense_linear} and $T_{linear_ordering}$ is the theory that extends the root theory with the axiom enforcing density between points, $T_{dense_partial}$. Comparing the set difference of the profiles of $T_{dense_partial}$ and $T_{linear_ordering}$, we see that it is equivalent to the difference between those theories.

3. Relationships between Hierarchies

Theories within the same hierarchy have the same non-logical lexicon; if we wish to consider the different relationships between theories in different hierarchies, we consider two cases – either the non-logical lexicon of one theory is an expansion of the non-logical lexicon of the other theory ($\Sigma(T_i) \subset \Sigma(T_j)$) or the non-logical lexicons of the two theories are disjoint ($\Sigma(T_i) \cap \Sigma(T_j) = \emptyset$). In this section, we explore these two possibilities and discuss their roles in providing an adequate notion of ontology modularity.

3.1. Hierarchies and Conservative Extensions

The simplest relationship between theories that are in different hierarchies is that of extension. In particular, the notion of conservative extension (see Definition 3) has played a key role in the study of modular ontologies [18,9].

The following theorem shows that we do not speak of one hierarchy \mathbb{H}_1 being a non-conservative extension of another hierarchy \mathbb{H}_2 , since we can always add new theories to \mathbb{H}_2 that are conservatively extended by theories in \mathbb{H}_1 .

Theorem 4 *Suppose T_1, T_2 are theories that are in different hierarchies such that $\Sigma(T_1) \subset \Sigma(T_2)$.*

If T_2 is a non-conservative extension of T_1 , then there exists a theory T_3 such that

- *T_2 is a conservative extension of T_3 , and*
- *T_3 is compatible with the hierarchy of T_1 , i.e., $\Sigma(T_3) = \Sigma(T_1)$.*

Proof: Suppose T_2 is a non-conservative extension of T_1 and $\Sigma(T_1) \subset \Sigma(T_2)$.

If

$$\Omega = \{\omega : \omega \in \mathcal{L}(T_1), T_1 \not\models \omega, T_2 \models \omega\}$$

then $T_3 = T_1 \cup \Omega$ is a non-conservative extension of T_1 with $\Sigma(T_1) = \Sigma(T_3)$, and T_2 is a conservative extension of $T_1 \cup \Omega$. \square

If we include the new theory T_3 in the hierarchy containing T_1 , the non-conservative extensions is limited to this hierarchy.

An example of a modular ontology that is organized through both conservative and non-conservative extensions is the PSL Ontology [12]. The theories that constitute PSL Outer Core are shown in Figure 3⁵. The theory $T_{occtree}$ is a non-conservative extension of $T_{pslcore}$ which also expands the non-logical lexicon. Following Theorem 4, there exists a theory T_{aoa} that is a non-conservative extension of $T_{pslcore}$ within the same hierarchy such that $T_{occtree}$ is a conservative extension of T_{aoa} . Similarly, the theory T_{atomic} is a non-conservative extension of $T_{subactivity}$ which also expands the non-logical lexicon. The theory T_{sublat} is a non-conservative extension of $T_{subactivity}$ within the same hierarchy such that T_{atomic} is a conservative extension of T_{sublat} .

In the case of using \mathcal{E} -connections to decompose a large ontology into modules (a collection of axioms), each module encapsulates some terms of the original ontology.

⁵The CLIF axiomatization of these theories can be found at <http://stl.mie.utoronto.ca/colore/process/>

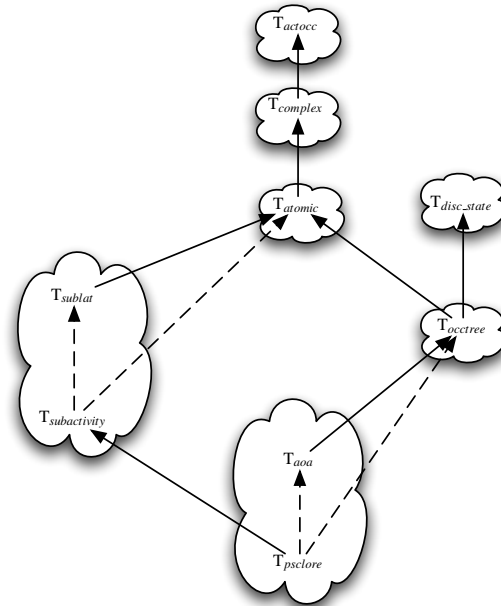


Figure 3. Relationships between the theories in PSL Outer Core. Clouds indicate different hierarchies. Dashed lines denote non-conservative extension while solid lines denote conservative extensions.

In [8] the definition of semantic encapsulation is given as a component that preserves a basic set of entailments of a term in an ontology. This leads to the partitioning of a large ontology into a collection of modules that are conservative extensions of one another. The relationship of conservative extensions between modules ensures that each module can be reused independent of the rest while retaining the original semantics of its contained terms. The major limitation of using \mathcal{E} -connections is that the use of such modules as a means of refining an ontology (non-conservative extensions of a theory) becomes impossible.

Using only the relationship of conservative extension does not give us a sufficient characterization of the granularity of a module. PSL Outer Core gives us examples of theories that are conservative extensions, yet we can decompose them into smaller theories through reducibility. Furthermore, there are also conservative extensions that are irrelevant to modularity. For example, the Peano Arithmetic is a conservative extension of the theory of linear orderings, yet we would not say that the latter theory forms a module within the former theory.

3.2. Interpretability between Theories

Ontologies whose non-logical lexicons are expansions of other non-logical lexicons can be compared using the notions of satisfiability, extension, and independence. More difficult is to compare ontologies that are axiomatized using different non-logical lexicons; in such cases, we need to determine whether or not the non-logical lexicon of one ontology can be interpreted in the non-logical lexicon of the other ontology. For this purpose

we assume that the two theories use disjoint non-logical lexicon; if they are not disjoint a simple renaming of one of the lexicons will do the trick. In this section, we consider metatheoretic relationships between theories based on the notion of relative interpretation, and we also discuss how they can support an approach to ontology modularity through the organization of ontologies into a repository.

3.2.1. Relative Interpretation

The notion of interpretability between theories is widely used within mathematical logic and applications of ontologies for semantic integration ([22]). We will adopt the following definition from [5]:

Definition 15 An interpretation π of the theory T_1 with signature $\Sigma(T_1)$ into a theory T_2 with signature $\Sigma(T_2)$ is a function on the set of non-logical symbols of $\Sigma(T_1)$ and formulae in $\mathcal{L}(T_1)$ such that

1. π assigns to \forall a formula π_{\forall} of $\mathcal{L}(T_2)$ in which at most the variable v_1 occurs free, such that

$$T_2 \models (\exists v_1) \pi_{\forall}$$

2. π assigns to each n -place relation symbol P a formula π_P of $\mathcal{L}(T_2)$ in which at most the variables v_1, \dots, v_n occur free.
3. π assigns to each n -place function symbol f a formula π_f of $\mathcal{L}(T_1)$ in which at most the variables v_1, \dots, v_n, v_{n+1} occur free, such that

$$T_1 \models (\forall v_1, \dots, v_n) \pi_{\forall}(v_1) \wedge \dots \wedge \pi_{\forall}(v_n)$$

$$\supset (\exists x) (\pi_{\forall}(x) \wedge ((\forall v_{n+1}) (\pi_f(v_1, \dots, v_{n+1}) \equiv (v_{n+1} = x))))$$

4. for any atomic sentence $\sigma \in \mathcal{L}(T_1)$ with relation symbol P , $\pi(\sigma) = \pi(P)$;
5. for any sentence $\sigma \in \mathcal{L}(T_1)$,
 $\pi(\neg\sigma) = \neg\pi(\sigma)$;
6. for any sentences $\sigma, \tau \in \mathcal{L}(T_1)$,
 $\pi(\sigma \rightarrow \tau) = \pi(\sigma) \rightarrow \pi(\tau)$;
7. for any sentence $\sigma \in \mathcal{L}(T_1)$,
 $\pi(\forall x \sigma) = \forall x \pi_{\forall} \rightarrow \pi(\sigma)$;
8. For any sentence $\sigma \in \mathcal{L}(T_1)$,

$$T_1 \models \sigma \Rightarrow T_2 \models \pi(\sigma)$$

Thus, the mapping π is an interpretation of T_1 if it preserves the theorems of T_1 and we say T_1 is interpretable in T_2 .

Definition 16 An interpretation π of a theory T_1 into a theory T_2 is faithful iff

$$T_1 \not\models \sigma \Rightarrow T_2 \not\models \pi(\sigma)$$

for any sentence $\sigma \in \mathcal{L}(T_1)$.

Thus, the mapping π is a faithful interpretation of T_1 if it preserves satisfiability with respect to T_1 . We will also refer to this by saying that T_1 is faithfully interpretable in T_2 . For example, the work in [11] shows that the PSL-Core theory within the PSL Ontology [12] is interpretable by Reiter's axiomatization of situation calculus, but that this is not a faithful interpretation, since there are sentences consistent with PSL-Core that are not consistent with situation calculus.

Definable equivalence is a generalization of the notion of logical equivalence to theories that do not have the same signature.

Definition 17 *Two theories T_1 and T_2 are definably equivalent iff T_1 is faithfully interpretable in T_2 and T_2 is faithfully interpretable in T_1 .*

For example, the theory of timepoints is definably equivalent to the theory of linear orderings. On the other hand, although the theory of partial orderings is interpretable in the theory of timepoints, these two theories are not definably equivalent, since the theory of timepoints is not interpretable in the theory of partial orderings.

Faithful interpretations are a generalization of the notion of conservative extension, leading to the following generalization of Theorem 4:

Theorem 5 *T_1 is faithfully interpretable in T_2 iff there is theory T_3 such that T_1 is definably equivalent to T_3 and T_2 is a conservative extension of T_3 .*

Proof: Suppose T_1 is definably equivalent to T_3 and T_2 is a conservative extension of T_3 .

By the definition of definable equivalence, we know that T_1 is faithfully interpretable in T_3 , so that there exists an interpretation π of T_1 into T_3 such that

$$T_1 \models \sigma \Leftrightarrow T_3 \models \pi(\sigma)$$

Since T_2 is a conservative extension of T_3 , we have

$$T_3 \models \sigma \Leftrightarrow T_2 \models \pi(\sigma)$$

and hence T_1 is faithfully interpretable in T_2 .

For the converse, suppose that T_1 is faithfully interpretable in T_2 with the interpretation π . Let

$$T_3 = \{\pi(\sigma) : \sigma \in T_1\}$$

Note that $\Sigma(T_3) = \Sigma(T_1)$, and that T_3 is definably equivalent to T_1 .

T_2 is a conservative extension of T_3 ; otherwise, there exists a sentence $\sigma \in \mathcal{L}(T_3)$ such that

$$T_3 \not\models \sigma \quad \text{and} \quad T_2 \models \sigma$$

However, since T_3 is definably equivalent to T_1 , this would mean that

$$T_1 \not\models \sigma$$

which contradicts the definable equivalence of T_1 and T_2 . \square

The use of relative interpretations between first-order axiomatized theories as a means to explicitly combine smaller theories was introduced by [1]. This approach has been implemented by the Interactive Mathematical Proof System (IMPS) [6], which is theorem prover that utilizes a repository of axiomatized mathematical theories linked to each other through relative interpretations using the little theories approach to mechanize traditional tools of classical mathematical reasoning [7]. The use of relative interpretations by IMPS provide the means to transport a theorem from the theory it was proved into any other theory linked with an interpretation. The IMPS repository is organized around the relative interpretations available between stored theories. Furthermore, IMPS guarantees the consistency of generated proofs based on the notion of relative consistency between theories. Within IMPS there is a set of theories deemed *foundational*, meaning they are regarded or known to be consistent. Since all proofs begin with a foundational theory and any theory developed from another is a conservative extension of the original theory, all theories developed are consistent relative to the original foundational theory [6]. Although the use and definitions of theory interpretations and relative consistency in IMPS are specific to the purpose of theorem proving, it nonetheless shows how such relationships can be utilized to relate and combine theories.

Nevertheless, since relative interpretation and definable equivalence are generalizations of the notion of extension, these relationships between theories suffer from the same drawbacks when attempting to decompose an ontology into modules. In particular, knowing that one theory T_1 is interpretable by a theory T_2 does not explicitly identify the subtheory of T_1 that is definably equivalent to T_2 .

3.2.2. Reducibility

Definable equivalence is a relationship between two theories; we can generalize this to a relationship among a set of theories. The basis for this approach is the model-theoretic notion of reducibility introduced in [13]. In this paper, we characterize the relationship between theories that corresponds to the model-theoretic notion of reducibility.

Definition 18 A theory T is *reducible* to a set of theories T_1, \dots, T_n iff

1. T faithfully interprets each theory T_i , and
2. $T_1 \cup \dots \cup T_n$ faithfully interprets T .

We will also refer to the set of theories T_1, \dots, T_n in the definition as the reduction of T in the repository.

It is easy to see that two definably equivalent theories are reducible to each other. For example, the theory of timepoints is reducible to the theory of linear ordering and vice versa.

A nontrivial example of reducibility can be seen with $T_{pslcore}$ within the PSL Ontology (see Figure 4). As discussed in [13], $T_{pslcore}$ is reducible to T_{linear} , $T_{partition}$, $T_{graph-incidence}$. This example also illustrates how the notion of reducibility leads to the decomposition of a theory that is treated as a module within a larger ontology, addressing the issue of the granularity of modules.

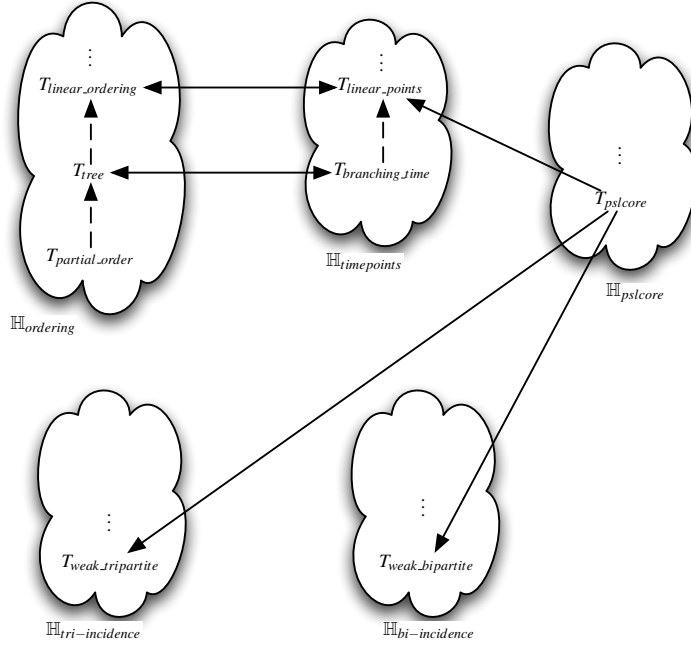


Figure 4. A depiction of the reducibility of $T_{pslcore}$ to other core theories. Clouds indicate hierarchies; only the root theory of each hierarchy and other relevant theories are shown. Dashed lines denote non-conservative extension within a hierarchy and solid lines denote reducibility of theories between hierarchies.

Lemma 6 Let T_1, \dots, T_n be a set of theories such that $\Sigma(T_i) \cap \Sigma(T_j) = \emptyset$ for all $1 \leq i, j \leq n, i \neq j$.

If T faithfully interprets each T_1, \dots, T_n , then T faithfully interprets $T_1 \cup \dots \cup T_n$.

Proof: Since the signatures of the theories are disjoint, we can construct an interpretation π of $T_1 \cup \dots \cup T_n$ in T by the union of the interpretations π_i of each T_i in T . \square

Theorem 6 Let T_1, \dots, T_n be a set of theories such that $\Sigma(T_i) \cap \Sigma(T_j) = \emptyset$ for all $1 \leq i, j \leq n, i \neq j$.

A theory T is reducible to T_1, \dots, T_n iff T is definably equivalent to

$$T_1 \cup \dots \cup T_n$$

Proof: Let T_1, \dots, T_n be a set of theories such that $\Sigma(T_i) \cap \Sigma(T_j) = \emptyset$ for all $1 \leq i, j \leq n, i \neq j$.

If T is reducible to T_1, \dots, T_n , then $T_1 \cup \dots \cup T_n$ (faithfully) interprets T ; by Lemma 6 T (faithfully) interprets $T_1 \cup \dots \cup T_n$, so it is easy to see that T is definably equivalent to $T_1 \cup \dots \cup T_n$.

Suppose T is not reducible to a set of theories T_1, \dots, T_n but T is definably equivalent to $T_1 \cup \dots \cup T_n$. Either T does not faithfully interpret some T_i or $T_1 \cup \dots \cup T_n$ does not faithfully interpret T . In the first case, there exists a sentence $\sigma \in \mathcal{L}(T_i)$ and an interpretation π such that

$$T_i \not\models \sigma, \quad T \models \pi(\sigma)$$

However, T is definably equivalent to $T_1 \cup \dots \cup T_n$, so that any sentence entailed by T is also entailed by $T_1 \cup \dots \cup T_n$, and since $\sigma \in \mathcal{L}(T_i)$, we must have $T_i \models \sigma$, which violates the assumption that T does not faithfully interpret some T_i . A similar argument applies for the other case, thus contradicting the assumption that T is not reducible to a set of theories T_1, \dots, T_n . \square

Corollary 1 *If T_1 is definably equivalent to T_2 , then T_1 is reducible to T_2 .*

In the following, we will see that reducibility is the metatheoretic relation we need for both the organization of ontology repositories and a definition of modules within an ontology.

3.3. Ordering Hierarchies by Faithful Interpretation

In the same way that we defined a hierarchy to be a set of theories that are partially ordered by extension, we can use faithful interpretation and reducibility to define a quasi-ordering over hierarchies.

Definition 19 *Let $\mathbb{H}_i = \langle \mathcal{H}_i, \leq \rangle$ and $\mathbb{H}_j = \langle \mathcal{H}_j, \leq \rangle$ be hierarchies.*

$\mathbb{H}_i \preceq \mathbb{H}_j$ *iff each root theory in \mathcal{H}_j faithfully interprets a theory T that is compatible with \mathcal{H}_i .*

For example⁶, we have the following relationships:

$$\mathbb{H}_{\text{between}} \preceq \mathbb{H}_{\text{ordering}} \preceq \mathbb{H}_{\text{timepoints}} \preceq \mathbb{H}_{\text{pslcore}}$$

Since the \preceq relation on hierarchies plays a key role, we show that it is indeed an ordering relation on the set of hierarchies.

Lemma 7 *If T_1 is faithfully interpretable in T_2 and T_2 is faithfully interpretable in T_3 , then T_1 is faithfully interpretable in T_3 .*

Proof: T_1 is faithfully interpretable in T_2 iff there exists an interpretation π_1 such that

$$T_1 \models \sigma \Leftrightarrow T_2 \models \pi_1(\sigma)$$

⁶We adopt a naming convention for hierarchies similar to the one that we use for theories. Each hierarchy is associated with a URI such that the subscript in the hierarchy name is the unqualified name in the URI. For example, the theories in the hierarchy $\mathbb{H}_{\text{ordering}}$ can be found at <http://stl.mie.utoronto.ca/colore/ordering/>.

Also note that the hierarchy $\mathbb{H}_{\text{ordering}}$ is depicted in Figure 1 and that the hierarchy $\mathbb{H}_{\text{timepoints}}$ is depicted in Figure 2.

T_2 is faithfully interpretable in T_3 iff there exists an interpretation π_2 such that

$$T_2 \models \sigma \Leftrightarrow T_3 \models \pi_2(\sigma)$$

Combining these gives us

$$T_1 \models \sigma \Leftrightarrow T_3 \models \pi_2\pi_1(\sigma)$$

so that $\pi_2\pi_1$ is a faithful interpretation of T_1 in T_3 . \square

Lemma 8 *If T_1 is faithfully interpretable in T_2 and $T_1 < T_3$, then T_3 is faithfully interpretable in some extension of T_2 .*

Proof: T_1 is faithfully interpretable in T_2 iff there is an interpretation π such that

$$T_1 \models \omega \Leftrightarrow T_2 \models \pi(\omega)$$

Suppose $T_1 < T_3$ (so that T_1 and T_3 are in the same hierarchy) and let

$$\Omega = \{\omega : T_3 \models \omega, T_1 \not\models \omega\}$$

$$\pi(\Omega) = \{\pi(\omega) : \omega \in \Omega\}$$

We can specify a new theory

$$T_4 = T_2 \cup \pi(\Omega)$$

such that

$$T_3 \models \omega \Leftrightarrow T_4 \models \pi(\omega)$$

so that T_3 is faithfully interpretable in an extension of T_2 . \square

Lemma 9 *The \preceq relation is a quasiordering on the set of hierarchies.*

Proof: \preceq is reflexive because the root theory of any hierarchy is reducible to itself.

Suppose

$$\mathbb{H}_1 \preceq \mathbb{H}_2, \mathbb{H}_2 \preceq \mathbb{H}_3$$

Each root theory T_{3i}^{root} faithfully interprets some theory T_{2i} that is compatible with \mathbb{H}_2 .

There exists a root theory in \mathbb{H}_2 such that $T_{2i}^{root} < T_{2i}$ and this root theory faithfully interprets some theory T_{1i} compatible with \mathbb{H}_1 .

By Lemma 8, T_{2i} faithfully interprets an extension T_{11} of T_{2i} .

By Lemma 7, the root theory T_{3i}^{root} faithfully interprets some theory T_{11} compatible with \mathbb{H}_1 , and hence $\mathbb{H}_1 \preceq \mathbb{H}_3$. \square

From the example presented at the beginning of this section, we can already see that this ordering relation alone will not be sufficient as the basis for an approach to modularity. The theories of betweenness relations in $\mathbb{H}_{between}$ are intuitively the weakest, since they can be faithfully interpreted in the other theories in the example. However, theories in $\mathbb{H}_{ordering}$ are not reducible to theories in $\mathbb{H}_{between}$. As a result, models of the theories in $\mathbb{H}_{between}$ cannot be used to construct models of the theories in the other hierarchies such as orderings, so that the theories in $\mathbb{H}_{between}$ fail to meet the intuition that the weakest theories form modules that are the building blocks of other theories.

3.4. What is an Ontology Repository?

In this section, we investigate the use of reducibility as a means of organizing the hierarchies in a repository. Recall that even though an actual repository can only contain ontologies which are concrete sets of axioms, we will again discuss the more abstract *repository of theories*.

Definition 20 Let $\mathbb{H}_1, \dots, \mathbb{H}_n$ be a finite set of hierarchies.

A repository $\mathbb{R} = \langle \mathcal{R}, \sqsubseteq \rangle$ is a partially ordered set where

- $\mathcal{R} = \{\mathbb{H}_1, \dots, \mathbb{H}_n\}$;
- $\mathbb{H}_i \sqsubseteq \mathbb{H}_j$ iff each root theory in \mathbb{H}_j has a reduction that contains a theory T that is compatible with \mathbb{H}_i .

For example, we can see that $\mathbb{H}_{ordering} \sqsubseteq \mathbb{H}_{timepoints}$, since the root theory in $\mathbb{H}_{timepoints}$ is definably equivalent to the theory $T_{linear_ordering}$ in $\mathbb{H}_{ordering}$. On the other hand, $\mathbb{H}_{timepoints} \not\sqsubseteq \mathbb{H}_{ordering}$, since the root theory for $\mathbb{H}_{ordering}$ (which is $T_{partial_ordering}$) is not reducible to any theory in $\mathbb{H}_{timepoints}$.

The reducibility of $T_{pslcore}$ leads to the following relationships among hierarchies (which is also depicted in Figure 4):

$$\mathbb{H}_{ordering} \sqsubseteq \mathbb{H}_{timepoints} \sqsubseteq \mathbb{H}_{pslcore}$$

$$\mathbb{H}_{tri_incidence} \sqsubseteq \mathbb{H}_{pslcore}$$

$$\mathbb{H}_{bi_incidence} \sqsubseteq \mathbb{H}_{pslcore}$$

It is important to note that the \sqsubseteq on hierarchies is distinct from this ordering based on faithful interpretation. For example, theories of betweenness relations are faithfully interpretable in theories of orderings relations, but they are not reducible to theories of orderings. Consequently, the hierarchies containing these theories would be incomparable in the repository with respect to the \sqsubseteq ordering.

Nevertheless, we do have the following relationships between the two orderings, which follows easily from the definition of reducibility:

Lemma 10 Let $\mathbb{H}_i = \langle \mathcal{H}_i, \leq \rangle$ and $\mathbb{H}_j = \langle \mathcal{H}_j, \leq \rangle$ be hierarchies.

$$\mathbb{H}_i \sqsubseteq \mathbb{H}_j \Rightarrow \mathbb{H}_i \preceq \mathbb{H}_j$$

We can use earlier results about the \preceq to establish properties of the \sqsubseteq ordering over hierarchies.

Theorem 7 *A repository $\mathbb{R} = \langle \mathcal{R}, \sqsubseteq \rangle$ is a quasiordering on the set of hierarchies.*

Proof: \sqsubseteq is reflexive because the root theory of any hierarchy is reducible to itself.

Now suppose $\mathbb{H}_i \sqsubseteq \mathbb{H}_j$ and $\mathbb{H}_j \sqsubseteq \mathbb{H}_k$.

By Lemma 10, we know $\mathbb{H}_i \preceq \mathbb{H}_j$ and $\mathbb{H}_j \preceq \mathbb{H}_k$.

By Lemma 7, we know $\mathbb{H}_i \preceq \mathbb{H}_k$.

Thus, each root theory in \mathbb{H}_k faithfully interprets a theory compatible with \mathbb{H}_i .

Since $\mathbb{H}_j \sqsubseteq \mathbb{H}_k$, there exist theories $S_1, \dots, S_j, \dots, S_n$ such that $S_1 \cup \dots \cup S_j \cup \dots \cup S_n$ faithfully interprets a root theory S_k in \mathbb{H}_k .

Since $\mathbb{H}_i \sqsubseteq \mathbb{H}_j$, there exist theories $T_1, \dots, T_i, \dots, T_m$ such that $T_1 \cup \dots \cup T_i \cup \dots \cup T_m$ faithfully interprets a root theory S in \mathbb{H}_j .

By Lemma 8, $T_1 \cup \dots \cup T_i' \cup \dots \cup T_m$ faithfully interprets S_j in \mathbb{H}_j .

By Lemma 7, $S_1 \cup \dots \cup (T_1 \cup \dots \cup T_i' \cup \dots \cup T_m) \cup \dots \cup S_n$ faithfully interprets S_k .

By Theorem 6, S_k is reducible to this set of theories and consequently $\mathbb{H}_i \sqsubseteq \mathbb{H}_j$.

Thus, \sqsubseteq is transitive. \square

3.5. Core Hierarchies

Since we are dealing with repositories that contain a finite set of hierarchies, we are guaranteed that the partial ordering \sqsubseteq has minimal elements.

Definition 21 *A hierarchy $\mathbb{C} = \langle \mathcal{C}, \leq \rangle$ is a core hierarchy iff it is a minimal hierarchy in the repository $\mathbb{R} = \langle \mathcal{R}, \sqsubseteq \rangle$.*

A theory T is a core theory iff it is compatible with a core hierarchy.

The core repository is the set of core hierarchies within a repository \mathbb{R} .

A complex hierarchy $\mathbb{H} = \langle \mathcal{H}, \leq \rangle$ is a hierarchy which is not minimal in the repository $\langle \mathcal{R}, \sqsubseteq \rangle$.

A theory T is a complex theory iff it is compatible with a complex hierarchy.

Using the examples of the preceding section, $\mathbb{H}_{ordering}$ is a core hierarchy and all theories that axiomatize different classes of orderings are core theories. On the other hand, $\mathbb{H}_{timepoint}$ and $\mathbb{H}_{pslcore}$ are complex hierarchies. In particular, $T_{pslcore}$ is a complex theory; as we saw in Figure 4, this theory is reducible to a set of core theories. The next theorem shows that this property holds for all complex theories:

Theorem 8 *If $\mathbb{H} = \langle \mathcal{H}, \leq \rangle$ is a complex hierarchy, then each theory in \mathcal{H} is reducible to a set of core theories.*

Proof: Suppose that $\mathbb{R} = \langle \mathcal{R}, \sqsubseteq \rangle$ is a repository such that $\mathcal{R} = \{\mathcal{H}_1, \dots, \mathcal{H}_n\}$.

Let \mathbb{H} be a complex hierarchy in \mathcal{R} , so that it is not minimal in the \sqsubseteq quasiordering (following Theorem 7). Because the repository is finite, and all finite quasiorderings are well-founded, there exist core hierarchies $\mathbb{C}_1, \dots, \mathbb{C}_n$ with $\mathbb{C}_i = \langle \mathcal{C}_i, \leq \rangle$ such that

$$\mathbb{C}_i \sqsubseteq \mathbb{H}$$

By the definition of \sqsubseteq , each root theory T in \mathbb{H} is reducible to

$$T_1, \dots, T_n$$

where $T_i \in \mathcal{C}_i$. Since core hierarchies are minimal in the \sqsubseteq quasiordering, all of these theories T_i are core theories. \square

Theorem 8 shows us that core theories are intuitively the building blocks of complex theories, analogous to the way that trunk theories are the building blocks of other theories in the same hierarchy. These insights lead us to a characterization of modularity.

3.6. Relationships between Theories

Thus far in the paper, we have used three different metatheoretic relationships to organize a first-order ontology repository – non-conservative extension (within a hierarchy), faithful interpretation, and reducibility (between different hierarchies). In each case, we can specify a partial ordering over the set of theories.

There have been several other approaches that have used similar relationships. In his lattice of theories, Sowa [23] describes a partial ordering over all first-order theories based on the notion of generalization and specialization. The top element of the lattice is the set of first-order tautologies, and each theory below the top is an extension of the ones above it. Sowa informally proposes four operations between theories within the lattice – contraction, expansion, revision, and analogy. Although he does not propose formal definitions for these operations, it appears that analogy could be defined with respect to our notion of faithful interpretation.

In contrast to the approach in this paper, Sowa’s lattice of theories contains *all* possible first-order theories in *all* possible signatures. Most importantly, Sowa does not distinguish between conservative and non-conservative extension within this lattice of theories; consequently, no distinction can be made between the ordering over theories within a hierarchy and the ordering over theories between hierarchies.

Another approach that utilizes metatheoretic relationships among ontologies is HETS [20]. Using category theory as a mathematical basis, HETS defines the notions of conservative extension and definable interpretation between theories. In this sense, HETS could provide an alternative architecture to implement the ideas in this paper, although HETS does not have a relationship that is equivalent to reducibility. In addition, the notion of an ontology module within HETS [19] emphasizes the role of conservative extension.

4. Two Characterizations of Modularity

Repositories as we defined them lend themselves to two notions of modularity, one defined using similarities and another defined using reducibility. We conclude this section by showing how they complement each other in modularizing an ontology.

Notice that the resulting modules are not linked to a specific axiomatization. In fact, we do not distinguish between two equivalent theories with different axiomatizations. Likewise, we do not distinguish two modules with different axiomatizations as long as they are logically equivalent.

4.1. Similarity

As we saw in Theorem 1, any ontology that is a non-root core theory in a core hierarchy has a unique profile consisting of a set of trunk theories. This set provides a natural modularization of the ontology: each trunk theory is a module that encapsulates an ontological design choice that the core theory shares with another theory in the same core hierarchy. All other theories in the hierarchy are combinations of the trunk theories, and additional modules arise to capture the similarities between different combinations.

Of course, the trunk theories are combined in a non-conservative way, which differs from typical approaches to modularity such as [9]. Nevertheless, the use of trunk theories for the notion of primitive module fits quite well with the intuition that they capture the smallest possible theory which can be shared and reused among multiple ontologies.

4.2. Reducibility

If an ontology is not a core theory in the repository, then the notion of trunk theory alone is insufficient to specify the modules within the ontology. In this case, we can use the notion of reducibility to decompose the ontology into modules that are definably equivalent to core theories within the repository.

Lemma 11 *If a theory T is reducible to a set of core theories in a repository \mathbb{R} , there exist core theories S_1, \dots, S_n such that T is reducible to S_1, \dots, S_n and each S_i belongs to a different core hierarchy.*

Proof: Suppose that there exist multiple theories S_{11}, S_{12} within the same core hierarchy \mathbb{C}_1 such that T is reducible to $S_1, \dots, S_{11}, S_{12}, \dots, S_n$.

By Theorem 6, T is definably equivalent to $S_1 \cup \dots \cup S_{11} \cup S_{12} \cup \dots \cup S_n$. In particular, T faithfully interprets both theories:

$$T \models \sigma \Leftrightarrow S_{11} \models \pi_1(\sigma)$$

$$T \models \sigma \Leftrightarrow S_{12} \models \pi_1(\sigma)$$

Since we also have

$$S_1 \cup \dots \cup S_{11} \cup S_{12} \cup \dots \cup S_n \models \sigma \Leftrightarrow T \models \pi(\sigma)$$

we know that S_{11} and S_{12} are logically equivalent. \square

If the repository contains multiple equivalent core hierarchies, then the reduction will contain multiple definably equivalent core theories, and hence there will exist multiple reductions that contain different sets of core theories.

Definition 22 *A minimal reduction of a theory T in a repository \mathbb{R} is a set of theories S_1, \dots, S_n such that S_i is not definably equivalent to S_j for any $1 \leq i, j \leq n$ such that $i \neq j$.*

The key result of this section shows how we can use the reduction of a complex theory to decompose it into a set of subtheories that are irreducible modules.

Theorem 9 *If S_1, \dots, S_n is a minimal reduction of a theory T in a repository \mathbb{R} , then there exist subtheories T_1, \dots, T_n of T such that*

- $T \models T_i$;
- T_i is definably equivalent to S_i .

Proof: If S_1, \dots, S_n is a minimal reduction of a theory T in a repository \mathbb{R} , then T is definably equivalent to $S_1 \cup \dots \cup S_n$ by Theorem 6.

There exists an interpretation π of $S_1 \cup \dots \cup S_n$ into T , so that for any $\sigma \in \mathcal{L}(S_1 \cup \dots \cup S_n)$,

$$S_1 \cup \dots \cup S_n \models \sigma \Leftrightarrow T \models \pi(\sigma)$$

For each S_i , there exists an interpretation π_i of T into S_i , so that for any $\sigma \in \mathcal{L}(T)$

$$T \models \sigma \Leftrightarrow S_i \models \pi_i(\sigma)$$

Let

$$T_i = \{\sigma : S_i \models \pi_i(\sigma), T \models \sigma\}$$

By the definition of reducibility, T faithfully interprets S_i , so that we have

$$S_i \models \sigma \Leftrightarrow T \models \pi(\sigma)$$

so that T_i is definably equivalent to S_i . \square

We will refer to the set of subtheories T_1, \dots, T_n as the modularization of the theory T in \mathbb{R} . The subtheory $(T \setminus \bigcup T_i)$ is referred to as the residue of T in the modularization.

Note that even if there exist multiple reductions that contain different sets of core theories, each reduction leads to the same modularization, since the different theories in the reductions are definably equivalent. As we saw with the notion of similarity, we also need to draw attention to the fact that the modularization is a relationship between subtheories of T and not simply between subsets of axioms of T .

4.3. Synthesis

The two notions of module presented in this section are compared in Table 1. Together they provide complementary approaches to both ontology modularity as well as ontology repositories. Any ontology can first be decomposed by finding the set of core theories that constitute its reduction. Each of these core theories can then be decomposed into the set of trunk theories in its profile. The first modularization step yields a set of theories that characterize the models of the ontology up to elementary equivalence, while the second modularization step supports a characterization of sharability and reuse with other ontologies.

<i>Relationship between Theories</i>	<i>Set of Theories</i>	<i>Atomic Module</i>	<i>Modularization</i>
non-conservative extension	hierarchy	trunk theory	profile
reducibility	repository	core theory	reduction

Table 1. Comparison between the two notions of modularity discussed in this paper.

This approach also addresses the problem of the potentially exponential number of modules, discussed in [3]. If we consider trunk theories in a hierarchy and core theories within the reduction of a theory to be the building blocks of ontologies, then the number of such modules is bounded by the number of axioms within the theory. Of course, this can lead to an exponential set of theories in the repository if we explicitly construct all possible combinations of trunk theories and core theories. In the remainder of this paper, we investigate a series of procedures which decompose any ontology into a set of core theories, and which identify the set of trunk theories which are equivalent to a core theory. The only theories that we require to explicitly be included within the repository are those required by core hierarchies being closed and atomistic.

5. Using the Repository

The applications of our notion of ontology repository revolve around two perspectives on the intuitions underlying the notion of “module”. In the first perspective, we decompose a first-order ontology into a set of disjoint subtheories which intuitively correspond to the modules of the ontology. The second perspective considers each trunk theory within a core hierarchy to be a module that axiomatizes a distinct ontological design choice within the setting of that core hierarchy. This notion of module is used to determine the relationships between theories with respect to the similarities and differences among them. The two perspectives on modularity motivate several scenarios that typify the application of an ontology repository in supporting the modularization of ontologies. The remainder of this paper will present semi-automated procedures that address these scenarios.

In the first scenario, we begin with a new ontology and determine whether or not it is reducible to a set of core theories within the repository. This captures the first perspective on modularity; as we will show in the *Decomp Procedure* (Section 6.4), reducibility can be used to identify those subtheories of the new ontology which are its modules. If a new ontology is not reducible, then there are two possibilities. In the optimistic case, the ontology is reducible to a set of theories which are extensions of core theories within the repository; once identified, the associated core hierarchies need to be updated with the new core theories. In the pessimistic case, the new ontology is a theory in a new core hierarchy, albeit one in which we have little understanding as to the intended models of its constituent theories. This case is essentially one of ontology design insofar as the verification of the ontology requires new representation theorems (since the characterization of the models cannot be reduced to the models of any existing ontologies in the repository).

The second scenario addresses an auxiliary theme of this paper, namely that ontology repositories are dynamic ontology design artefacts, and as such need to have the capability of being updated with new theories. The *Update Procedure* (Section 7.1) specifies exactly how one updates an existing core hierarchy with a new core theory while

propagating the changes so that the resulting hierarchy is closed and atomistic: every non-root theory in the hierarchy is the union of a set of trunk theories of the hierarchy (compare Def. 11). The new core theory can either arise from the modularization procedure in the first scenario, or it can simply be a new theory that has been proposed within the research community.

Requiring all core hierarchies to be closed and atomistic allows us to easily determine the similarities and differences among various core theories within a single core hierarchy. As we have already seen, the similarities between core theories correspond to the ontological design choices that they share (i.e. the trunk theories in their shared profiles) and the differences correspond to the design choices for which the theories disagree. The third scenario generalizes this idea to complex hierarchies, that is, it uses the relationships between hierarchies to determine the similarities between complex theories within the same complex hierarchy.

These scenarios assume that we already have a repository with a set of (closed and atomistic) core hierarchies and maybe some complex hierarchies as well. However, one might ask how we actually can get to this stage. How can we initially populate the repository? In fact, this case can be reduced to the other two scenarios.

We first need to identify all hierarchies. By comparing the roots of any two hierarchies, we can decide which hierarchies will initially be core hierarchies. This is not covered by the procedures, but we can determine it by finding the similarity between any roots of two hierarchies.

We can start with one of the roots of each such hierarchy and successively add the other roots while ensuring that the hierarchy is closed and atomistic. Any non-root core theory can then be inserted using the Update Procedure. Once this process completes, all core hierarchies are in the repository and are closed and atomistic.

Finally, the non-core hierarchies are inserted one at a time using the Decomp Procedure, potentially creating additional core theories.

6. Modularizing an Ontology

In this section, we present a semi-automated procedure that uses the repository to decompose an ontology T into a set of irreducible modules T_1, \dots, T_n . The basis for this procedure is Theorem 9 – we first find a set of core theories which form the reduction of T , and then we specify the subtheories of T which are definably equivalent to the core theories.

6.1. Translation Definitions

Recall that a set of theories S_1, \dots, S_n is a reduction of T iff T is definably equivalent to $S_1 \cup \dots \cup S_n$. The first step in the modularization of T will be to recast the metatheoretic relationship of relative interpretation into an entailment problem.

Definition 23 Let T_0 be a theory with signature $\Sigma(T_0)$ and let T_1 be a theory with signature $\Sigma(T_1)$ such that $\Sigma(T_0) \cap \Sigma(T_1) = \emptyset$. Translation definitions for T_0 into T_1 are conservative definitions in $\Sigma(T_0) \cup \Sigma(T_1)$ of the form

$$\forall \bar{x} p_i(\bar{x}) \equiv \Phi(\bar{x})$$

where $p_i(\bar{x})$ is a relation symbol in $\Sigma(T_0)$ and $\Phi(\bar{x})$ is a formula in $\mathcal{L}(T_1)$.

Following [24], translation definitions can be considered to be an axiomatization of the interpretation of T_0 into T_1 .

Theorem 10 T_1 is interpretable in T_2 iff there exist a set of translation definitions Δ for T_1 into T_2 such that

$$T_2 \cup \Delta \models T_1$$

Proof: If T_1 is interpretable in T_2 , then there exists an interpretation π such that π assigns to each n-place relation symbol P a formula π_P of $\mathcal{L}(T_1)$ in which at most the variables v_1, \dots, v_n occur free [5].

It is easy to see that π_P is the antecedent of a translation definition and that the literal $P(v_1, \dots, v_n)$ is the consequent of a translation definition.

Conversely, suppose we are given a set of translation definitions Δ such that $T_2 \cup \Delta \models T_1$, that is,

$$T_1 \models \sigma \Rightarrow T_2 \cup \Delta \models \sigma$$

For each translation definition,

$$\forall \bar{x} p_i(\bar{x}) \equiv \Phi_i(\bar{x})$$

specify the mapping

$$\pi(p_i) = \Phi_i(\bar{x})$$

For any sentence $\sigma \in \mathcal{L}(T_1)$, we know that $\pi(\sigma)$ is logically equivalent to σ by substitution of the consequents of the translation definitions in Δ , so that

$$T_2 \cup \Delta \models \pi(\sigma)$$

Since the translation definitions are conservative, this is equivalent to

$$T_2 \models \pi(\sigma)$$

and hence π is an interpretation. \square

Theorem 11 T_1 is faithfully interpretable in T_2 iff there exist a set of translation definitions Δ for T_1 into T_2 such that $T_2 \cup \Delta$ is a conservative extension of T_1 .

Proof: Theorem 10 shows that $T_2 \cup \Delta$ is an extension of T_1 .

Let π be the interpretation associated with the translation definitions Δ .

$T_2 \cup \Delta$ is a conservative extension of T_1 iff for every sentence $\sigma \in \mathcal{L}(T_1)$,

$$T_2 \cup \Delta \models \sigma \Rightarrow T_1 \models \sigma$$

which holds iff

$$T_2 \models \pi(\sigma) \Rightarrow T_1 \models \sigma$$

which is equivalent to π being a faithful interpretation. \square

If we recall the definition of reducibility, we need to find a set of theories such that T is definably equivalent to $T_1 \cup \dots \cup T_n$. The definition of definable equivalence requires the solution of two distinct sets of entailment problems:

$$T \cup \Delta_i \models T_i$$

$$T_1 \cup \dots \cup T_n \cup \Pi \models T$$

where Δ_i is the set of translation definitions from T to the theory T_i , and Π is the set of translation definitions from the combined theory $T_1 \cup \dots \cup T_n$ to T .

The next lemma shows that the same translation definitions can be used for all theories that are stronger than some theory in the same hierarchy.

Lemma 12 Let π be an interpretation of the theory T_1 in the hierarchy \mathbb{H}_1 into the theory T_2 in the hierarchy \mathbb{H}_2 . Let Δ be the set of translation definitions for π .

If

$$T_1 \cup \Delta \models T_2$$

and $T_1 < T_{11}$, then \mathbb{H}_2 can be extended with a theory T_{21} such that $T_2 < T_{21}$ and

$$T_{11} \cup \Delta \models T_{21}$$

Proof: It is easy to see that

$$(T_{21} \setminus T_2) = \{\pi(\sigma) : T_{11} \models \sigma, T_2 \not\models \pi(\sigma), T_{21} \models \pi(\sigma)\}$$

\square

In general, translation definitions cannot be automatically generated for two arbitrary theories, since it is not decidable to determine if an interpretation between the two theories exists.

6.2. Translations of Axioms

Rather than use the translation definitions as additional axioms used to entail the sentences in another theory, we can also specify a theory in which predicates in one signature are replaced by their definitions in the other signature.

Definition 24 Let Δ be the set of translation definitions Δ for the theory T_1 into the theory T_2 .

The translation T_1^r of T_1 is the set of sentences in $\mathcal{L}(T_2)$ in which each predicate in T_1 has been replaced by the consequent of its translation definition in Δ .

As an example, we can consider one of the theories in the reduction of the theory $T_{pslcore}$. The theory of linear orderings contains the sentence

$$(\forall x, y, z) leq(x, y) \wedge leq(y, z) \supset leq(x, z)$$

Using the translation definition into the theory $T_{pslcore}$:

$$(\forall x, y) leq(x, y) \equiv beforeEq(x, y)$$

the translation of the sentence is

$$(\forall x, y, z) beforeEq(x, y) \wedge beforeEq(y, z) \supset beforeEq(x, z)$$

This technique of using the translations of sentences will play a key role in the decomposition procedure introduced later in this section.

It is a straightforward corollary of Theorem 10 that the translation T_1^r of T_1 is logically equivalent to a subtheory of T_2 .

Corollary 2 Let Δ be the set of translation definitions Δ for the theory T_1 into the theory T_2 .

If T_1^r is the translation of T_1 , then

$$T_2 \models T_1^r$$

6.3. Finding Reducible Theories

The $FindTheory(\mathbb{R}, T, \mathcal{T})$ Procedure addresses the first set of entailment problems by finding the set \mathcal{T} of maximal core theories in the repository \mathbb{R} that are interpretable by the theory T .

Beginning with the root theory in a core hierarchy \mathbb{C} , the procedure searches⁷ through \mathbb{C} to find the maximal theories that are interpretable by T . If there are multiple maximal theories, the procedure returns the union of their sets of axioms.

⁷We assume the existence of three other algorithms related to partial orderings (see [2]). The first is $ChainDecomp(P)$ which outputs the set of chains for a poset P , and the second is $PosetSort(X, P)$, which constructs a poset P from an unordered set X . The third algorithm is $NextTheory(T, P)$ which returns the elements in the poset P that covers the element T .

It is important to notice that there are two steps within this procedure that require user interaction. First, someone must determine whether or not the root theory of a particular core hierarchy is interpretable by T . Second, the entailment problem in step 11 typically requires human guidance even with an automated theorem prover.

Procedure 1 $FindTheory(\mathbb{R}, T, \mathcal{T})$

Require: Core repository $\mathbb{R} = \langle \mathcal{R}, \sqsubseteq \rangle$, theory T .

Ensure: $T_i \in \mathcal{T}$ is a maximal theory in \mathbb{C}_i that is interpreted by T , for each core hierarchy $\mathbb{C}_i \in \mathbb{R}$.

$\mathcal{T} \leftarrow \emptyset$

for all $\mathbb{C}_i \in \mathcal{R}$ whose root theories are interpretable by T **do**

$\Delta_i \leftarrow$ translation definitions for T into theories in \mathbb{C}_i

$Chains_i \leftarrow ChainDecomp(\mathbb{C}_i)$

5: $Collect_i \leftarrow \emptyset$

for all $G_{ij} \in Chains_i$ **do**

$Candidate_{ij} \leftarrow \emptyset$

$T_{current} \leftarrow$ minimal theory in G_{ij}

$T_{max} \leftarrow$ maximal theory in G_{ij}

10: **while** $T_{current} \neq T_{max}$ **do**

if $T \cup \Delta_i \models T_{current}$ and $T \cup \Delta_i$ is a conservative extension of $T_{current}$ **then**

$Candidate_{ij} \leftarrow T_{current}$

$T_{current} \leftarrow NextTheory(G_{ij}, T_{current})$

else

15: $T_{current} \leftarrow T_{max}$

end if

end while

end for

$Collect_i \leftarrow \bigcup_j \{Candidate_{ij}\}$ (the set of candidate theories)

20: $PosetSort(Collect_i, P)$

$T_i \leftarrow$ set of axioms in the union of maximal theories in P

$\mathcal{T} \leftarrow \mathcal{T} \cup \{T_i\}$

end for

We turn now to the correctness of the $FindTheory$ Procedure.

By the following lemma, if a theory T_i in a core hierarchy \mathbb{C}_i is not interpretable by T , then no extension of it in \mathbb{C}_i is interpretable by T . In particular, if the root theory of \mathbb{C}_i is not interpretable by T , then no theory in \mathbb{C}_i is interpretable by T .

Lemma 13 Let $\mathbb{C} = \langle \mathcal{C}, \leq \rangle$ be a core hierarchy such that $T_1, T_2 \in \mathcal{C}$.

If T_2 is interpretable by T and $T_1 < T_2$, then T_1 is interpretable by T .

Proof: By Theorem 10 and Theorem 11, T_2 is interpretable by T iff there exist translation definitions Δ such that

$$T \cup \Delta \models T_2$$

and $T \cup \Delta$ is a conservative extension of T_2 .

Since $T_1 < T_2$, the definition of hierarchy gives us $T_2 \models T_1$ and hence

$$T \cup \Delta \models T_1$$

so that T_1 is interpretable by T . \square

Lemma 14 Let $\mathbb{C} = \langle \mathcal{C}, \leq \rangle$ be a closed core hierarchy.

For any theory T , there exists at most one maximal theory in \mathbb{C} that is interpretable by T .

Proof: Since \mathbb{C} is closed, it has a unique root theory.

By Lemma 13, if the root theory of \mathbb{C} is not interpretable by T , then no extension of the root theory is interpretable by T . Since all theories in \mathbb{C} are extensions of the root theory, no theories in \mathbb{C} are interpretable by T .

If there are multiple maximal theories T_1, \dots, T_n that are interpretable by T , then there exist translation definitions Δ such that

$$T \cup \Delta \models T_1 \cup \dots \cup T_n$$

so that $T_1 \cup \dots \cup T_n$ is a consistent extension of each T_i which is interpretable by T and is itself a theory in \mathbb{C} . \square

Theorem 12 Let $\mathbb{R} = \langle \mathcal{R}, \sqsubseteq \rangle$ be a repository, and let T be a theory such that $T \notin \mathcal{R}$.

If the $FindTheory(\mathbb{R}, T, \mathcal{T})$ Procedure terminates, then $T_i \in \mathcal{T}$ is a maximal theory in \mathbb{C}_i that is interpretable by T , for each core hierarchy $\mathbb{C}_i \in \mathbb{R}$.

Proof: The procedure has the following loop invariant:

L $Candidate_{ij}$ is a theory in a chain in the core hierarchy \mathbb{C}_i that is interpreted by T .

The root theory of \mathbb{C}_i is the unique minimal theory in each chain in \mathbb{C}_i , and by the condition in Line 1, the root theory is interpretable by T . The loop invariant is therefore satisfied by the preconditions of the procedure.

We next show that the loop invariant is preserved at the end of the iteration.

$Candidate_{ij}$ is updated in Line 12. By Lemma 12, the same translation definitions are used for all theories in the same hierarchy. By Theorem 10, Theorem 11, and the entailment problem in Line 11, if $T \cup \Delta_i \models T_{current}$, then $T_{current}$ is faithful interpretable by T .

If $T \cup \Delta_i \not\models T_{current}$, then by Theorem 10, $T_{current}$ is not interpretable by T . By Lemma 13, no extension of $T_{current}$ is interpretable by T . In this case, the loop terminates and $Candidate_{ij}$ is the maximal theory in the chain that is interpretable by T .

By the assignment in Line 18, $Collect_i$ is the set of maximal theories in each chain in \mathbb{C}_i . The external Procedure $PosetSort$ assigns to P the maximal elements in $Collect_i$. By Lemma 14, T_i is the unique maximal theory in \mathbb{C}_i that is interpretable by T .

Finally, we can see that if the entailment problem in Step 11 terminates, then the procedure terminates, since $PosetSort$ terminates and each core hierarchy is finite (and hence has minimal and maximal elements), so that the loops will eventually terminate. \square

6.4. Ontology Decomposition Procedure

In this section, we outline a semi-automated procedure for using the relationships among core theories and complex theories within the repository to decompose an ontology into irreducible modules.

Procedure 2 $Decomp(\mathbb{R}, T, \mathcal{M})$

Require: Core repository $\mathbb{R} = \langle \mathcal{R}, \sqsubseteq \rangle$, theory T .

Ensure: \mathcal{M} is the modularization of T

```

 $\mathcal{M} \leftarrow \emptyset$ 
 $\mathcal{T} \leftarrow \emptyset$ 
 $FindTheory(\mathbb{R}, T, \mathcal{T})$ 
 $\Pi \leftarrow$  translation definitions for theories in  $\mathcal{T}$  into  $T$ 
5: if  $\mathcal{T} \cup \Pi \models T$  and  $\mathcal{T} \cup \Pi$  is a conservative extension of  $T$  then
    for all  $S_i \in \mathcal{T}$  do
         $T_i \leftarrow \emptyset$ 
        for all  $\sigma \in S_i$  do
             $\sigma^r \leftarrow$  translation of  $\sigma$ 
10:          $T_i \leftarrow T_i \cup \{\sigma^r\}$ 
        end for
         $\mathcal{M} \leftarrow \mathcal{M} \cup \{T_i\}$ 
    end for
else
15: if  $\mathcal{T} \cup \Pi \cup T$  consistent then
         $T$  is reducible to an extension of theories in  $\mathcal{T}$ 
    else
         $T$  is not reducible to any set of theories in  $\mathbb{R}$ 
    end if
20: end if

```

The *FindTheory* Procedure finds a set of core theories which are interpretable by the theory T . In order to show that a subset of these theories constitutes a reduction of T , we need to show that the union of such theories interprets T .

As discussed above, a user is required to provide the translation definitions Π for the core theories $S_1 \cup \dots \cup S_n$ into T . If in fact we have $S_1 \cup \dots \cup S_n \cup \Pi \models T$, then by Theorem 8, we know that a modularization exists. Each module T_i of T is the subtheory of T which is equivalent to the core theory S_i ; we therefore determine which axioms within T are entailed by $S_i \cup \Pi$ for each core theory within the reduction of T .

If an ontology is not reducible to the set of theories in a repository, we can use the notion of weak interpretability [4] to determine whether it is possible to extend the repository to include new core theories in existing core hierarchies such that a reduction exists in the new repository.

Definition 25 A theory T_1 is weakly interpretable in theory T_2 iff there is a function on $\Sigma(T_1)$ such that for any sentence $\sigma \in \mathcal{L}(T_1)$,

$$T_1 \models \sigma \Rightarrow T_2 \not\models \neg\pi(\sigma)$$

The results of [4] show that a theory T_1 is weakly interpretable in a theory T_2 iff T_1 is interpretable in some consistent extension of T_2 .

Lemma 15 T is interpretable in some consistent extension of $S_1 \cup \dots \cup S_n$ iff there exist translation definitions Π such that

$$S_1 \cup \dots \cup S_n \cup \Pi \cup T$$

is consistent.

Proof: T is interpretable in some consistent extension of $S_1 \cup \dots \cup S_n$ iff there exist translation definitions Π such that

$$S'_1 \cup \dots \cup S'_n \cup \Pi \models T$$

where $S_i \subseteq S'_i$, from which it follows that $S_1 \cup \dots \cup S_n \cup \Pi \cup T$ is consistent. \square

This Lemma is the justification for lines 15-19 in the *Decomp* Procedure.

In general, the extension of the repository is an ontology design task. Under restricted conditions, however, we can identify new core theories. For example, suppose we have three core theories S_{11} , S_{12} , and S_{21} such that $S_{11} < S_{12}$ and

$$T \cup \Delta_1 \cup \Delta_2 \models S_{21} \cup S_{11}$$

$$S_{21} \cup S_{11} \cup \Pi \not\models T$$

but

$$S_{21} \cup S_{12} \cup \Pi \models T$$

$$T \cup \Delta_1 \cup \Delta_2 \not\models S_{21} \cup S_{12}$$

By Lemma 15, we know that there exists a theory S_{13} such that $S_{11} < S_{13}$ and

$$T \cup \Delta_1 \cup \Delta_2 \models S_{21} \cup S_{13}$$

$$S_{21} \cup S_{13} \cup \Pi \models T$$

and hence that T is reducible to $S_{21} \cup S_{13}$.

An additional problem is that if $S_1 \cup \dots \cup S_n \cup \Pi \not\models T$, multiple core theories may need to be extended, but we do not a priori know which ones need to be extended.

If a theory T is not reducible to any theories within the repository, then effectively T itself is the first theory in a new hierarchy of its own. We say that such a theory is under probation; as the etymology suggests, additional proofs are needed to determine the relationship between this new theory and existing theories within the repository. In particular, new representation theorems are required to establish the verification of the new theory.

Theorem 13 Let $\mathbb{R} = \langle \mathcal{R}, \sqsubseteq \rangle$ be a repository, and let T be a theory such that $T \notin \mathcal{R}$.
If T is reducible to a set of theories in the repository and the $Decomp(\mathbb{R}, T, \mathcal{M})$ Procedure terminate, then \mathcal{M} is a modularization of T .

Proof: By Theorem 12, $FindTheory(\mathbb{R}, T, \mathcal{T})$ returns a set of theories \mathcal{T} in \mathbb{R} that are interpretable by T .

There are three cases:

1. $\mathcal{T} \cup \Pi \models T$;
2. $\mathcal{T} \cup \Pi \not\models T$ but $\mathcal{T} \cup \Pi \cup T$ consistent;
3. $\mathcal{T} \cup \Pi \cup T$ is not consistent.

Case 1: (Line 5) By Theorem 6, T is reducible to the theories in \mathcal{T} , that is, \mathcal{T} is a reduction of T .

By Theorem 9, there exist subtheories T_1, \dots, T_n of T such that each T_i is definably equivalent to some $S_i \in \mathcal{T}$.

The outer loop (Lines 6-13) iterates through the theories in \mathcal{T} . The inner loop (Lines 8-11) iterates through the axioms in $S_i \in \mathcal{T}$. T_i^r is the set of translations of axioms in S_i . By Corollary 2, $T_i^r \subseteq T$ and T_i^r is definably equivalent to S_i .

The assignment in Line 12 collects the subtheories of T .

Case 2: (Line 15) By Lemma 15, T is reducible to extensions of the theories in \mathcal{T} .

Case 3: (Line 18) By Lemma 15, T is not reducible to any core theories in \mathbb{R} . \square

It is important to stress that $Decomp$ and $FindTheory$ are procedures and not algorithms, both because of the undecidable theorem proving and consistency-checking steps (Line 9 in $FindTheory$ and Lines 5 and 15 in $Decomp$), but also because of the user intervention required for the specification of translation definitions (Line 2 of $FindTheory$ and Line 4 of $Decomp$). In this sense, the procedures give practical guidance for the designers of modular ontologies.

7. Updating a Hierarchy

In the preceding section, we observed that if an ontology is not reducible to a set of theories already contained in a repository, then the repository can be extended to include new theories.

Here we present a semi-automated procedure that inserts a new theory into a closed, atomistic repository so that the resulting repository is still closed and atomistic. This procedure is intended to be used with the core hierarchies which are required to be closed and atomistic in our repository, but is not restricted to those. In general, this leads to the introduction of new trunk theories into the (core) hierarchy, as well as the refinement of some former trunk theories in the original (core) hierarchy.

Adding a theory T to an existing closed, atomistic hierarchy presumes, of course, that the signature of T is the same as the signature of the theories in the hierarchy, and that T is an extension of the root theory of the hierarchy. Procedures 3, 4, and 5 give the full details. The following notation is used throughout the procedures:

$root(\mathbb{C})$: Reference to the root theory of the closed, atomistic hierarchy \mathbb{C} .

$trunk(\mathbb{C})$: Reference to the set of trunk theories of the closed, atomistic hierarchy \mathbb{C} .

profile(T): Reference to the set of theories that constitute the profile of the theory T .
similarity(T_1, T_2): Returns a theory T that is the similarity of the theories T_1 and T_2 in the sense of Definition 7.
difference(T_1, T_2): Returns a theory T that is the difference between the theories T_1 and T_2 in the sense of Definitions 13 and 14.

When adding a theory T to a closed, atomistic hierarchy, there are three possibilities:

1. T is logically equivalent to an existing core theory;
2. T is logically equivalent to the union of a set of trunk theories in the hierarchy (i.e. the profile of T);
3. There does not exist a subset of the trunk theories whose union entails T .

Procedure 3 *UpdateHierarchy*(\mathbb{C}, T)

Require: A closed, atomistic hierarchy $\mathbb{C} = \langle \mathcal{C}, \leq \rangle$, and a theory T such that $\Sigma(\mathbb{C}) = \Sigma(T)$ and $T \models \text{root}(\mathbb{C})$.

Ensure: $\mathbb{C} = \langle \mathcal{C} \cup \{T\}, \leq \rangle$ is a closed, atomistic hierarchy.

UpdateProfile(\mathbb{C}, T)
 $\mathcal{C} \leftarrow \mathcal{C} \cup \{T\}$
CloseHierarchy(\mathbb{C}, T)

Procedure 4 *UpdateProfile*(\mathbb{C}, T)

Require: A closed, atomistic hierarchy $\mathbb{C} = \langle \mathcal{C}, \leq \rangle$, a theory T such that $\Sigma(\mathbb{C}) = \Sigma(T)$ and $T \models \text{root}(\mathbb{C})$.

Ensure: T has a profile in \mathbb{C} , \mathbb{C} is atomistic and the similarity between any two trunk theories is a root theory.

Covered $\leftarrow \emptyset$
 $T^d \leftarrow T$ (the difference between T and the currently covered trunk theories)
while $\exists T' \in \text{trunk}(\mathbb{C}) \setminus \text{Covered}$ with *similarity*(T', T^d) $\neq \text{root}(\mathbb{C})$ **do**
 $S \leftarrow \text{similarity}(T', T^d)$
5: *Covered* $\leftarrow \text{Covered} \cup \{S\}$
 $T^d \leftarrow \text{difference}(T^d, S)$
 $T'^d \leftarrow \text{difference}(T', S)$
 $\text{trunk}(\mathbb{C}) \leftarrow (\text{trunk}(\mathbb{C}) \setminus T') \cup \{S, T'^d\}$
 $\text{profile}(T') \leftarrow \{S, T'^d\}$
10: **for all** $T_i \in \mathbb{C}$ **do**
 if $T' \in \text{profile}(T_i)$ **then**
 $\text{profile}(T_i) \leftarrow (\text{profile}(T_i) \setminus T') \cup \text{profile}(T')$
 end if
 end for
15: **end while**
 $\text{trunk}(\mathbb{C}) \leftarrow \text{trunk}(\mathbb{C}) \cup \{T^d\}$
 $\text{profile}(T) \leftarrow \text{Covered} \cup \{T^d\}$

Procedure 5 *CloseHierarchy*(\mathbb{C}, T)

Require: An atomistic hierarchy $\mathbb{C} = \langle \mathcal{C}, \leq \rangle$ and a theory T such that $T \in \mathcal{C}$, T has a profile in \mathbb{C} and $\mathbb{C}' = \langle \mathcal{C} \setminus T, \leq \rangle$ is closed.

Ensure: $\mathbb{C} = \langle \mathcal{C} \cup \{T\}, \leq \rangle$ is a closed, atomistic hierarchy.

for all $T_i \in \mathcal{C}$ **do**

$profile(S) \leftarrow profile(T) \cap profile(T_i)$ (Similarity)

$S \leftarrow \bigcup_j S_j \in profile(S)$

if $S \notin \mathcal{C}$ **then**

5: $\mathcal{C} \leftarrow \mathcal{C} \cup S$

CloseHierarchy(\mathbb{C}, S)

end if

end for

In the first two cases, the profile of the new theory T consists entirely of already existing trunk theories. In the first case, the execution of *UpdateHierarchy* results in no change in the hierarchy. In the second case, the execution of the procedure amounts to identification of the profile of the new theory and the similarities and differences of the theory with other non-trunk theories.

The third case is more interesting, since it means that a new trunk theory must be added to the core hierarchy \mathbb{C} in order to maintain an atomistic hierarchy. This is handled by the *UpdateProfile* Procedure which inserts all necessary new trunk theories into the hierarchy and determines which (if any) existing trunk theories are no longer trunk theories. As a side effect, *UpdateProfile* determines the new profile for every existing theory in the hierarchy. In particular, the difference T^d of T with respect to the union of all trunk theories entailed by T , i.e. $\bigcup \{T' \mid T' \in trunk(\mathbb{C}) \wedge T \models T'\}$, is itself a theory in the same hierarchy by the definition of differences. Moreover, T_{comp} is independent of the other trunk theories that are in the profile of T . If there is no nontrivial similarity between T^d and the trunk theories (i.e. the similarity is the root theory), then T^d covers (see Def. 9) the root theory and is itself a new trunk theory. On the other hand, if T^d does have a nontrivial similarity S with one of the existing trunk theories T' , then T' no longer covers the root theory and hence it is no longer a trunk theory. In addition, the difference between T' and S is also a subtheory of T' and the difference between T^d and S is also a subtheory of T^d . Now S and both differences are trunk theories in the hierarchy – unless we can find further similarities with other trunk theories.

For the first two cases, that is, if no new trunk theory needs to be added, we only need to update the similarities to ensure closure. By Theorem 1 this can be achieved using the profiles alone (see *CloseHierarchy*) because the hierarchy is still atomistic. The *UpdateHierarchy* Procedure terminates in all three cases if all calls of *similarity* and *difference* terminate. In the following, we prove this property as well as correctness, i.e. that the resulting hierarchy $\mathbb{C} = \langle \mathcal{C} \cup T, \leq \rangle$ is closed (under similarities) and atomistic.

Lemma 16 Let $\mathbb{C} = \langle \mathbb{C}, \leq \rangle$ be a closed, atomistic hierarchy and let T be a theory such that $\Sigma(\mathbb{C}) = \Sigma(T)$ and $T \models \text{root}(\mathbb{C})$. If the *UpdateProfile*(\mathbb{C}, T) Procedure terminates, the resulting hierarchy $\mathbb{C}_{\text{new}} = \langle \mathbb{C}_{\text{new}}, \leq \rangle$ is atomistic and T has a profile in \mathbb{C}_{new} .

Proof: We begin by stating the loop invariants of the procedure:

L1 $\bigcup_i \text{Covered}_i \cup \{T^d\} = T$

L2 The similarity between any two trunk theories is the root theory.

L3 $\bigcup_i \text{Covered}_i$ has a profile in \mathbb{C} .

L4 \mathbb{C} is atomistic.

Let the preconditions of the procedure be ρ and let $T^d = T$ and $\text{Covered} = \emptyset$ as set before entering the loop. Then,

$\rho \Rightarrow L1$: Since $\bigcup_i \text{Covered}_i = \emptyset$ and $T^d = T$

$\rho \Rightarrow L2$: By Lemma 4

$\rho \Rightarrow L3$: The profile of $\bigcup_i \text{Covered}_i$ is the empty profile.

$\rho \Rightarrow L4$: Trivial.

Next we show that given the loop invariants hold before an iteration, they are preserved at the end of the iteration.

L1: Line 5 adds a theory S to the *Covered* and line 6 updates T^d to be the difference between the previous T^d and S . *L1* follows from the definitions of remainder and difference.

L2: Trunk theories are updated in line 8. The condition holds since $T' = S \cup \{T'^d\}$ and the similarity between S and T'^d is the root theory as T'^d is the difference between T' and S (line 7).

L3: The only theory that is added to *Covered* is S . Since S is added to the trunk theories in line 8, *Covered* has a profile in \mathbb{C} .

L4: The only way this condition can be broken is if a non-trunk theory is created that does not have a profile in \mathbb{C} . Line 8, is the only operation in the procedure that creates a non-trunk theory by removing T' from $\text{trunk}(\mathbb{C})$. Since $T' = S \cup \{T'^d\}$ and both S and T'^d are added to $\text{trunk}(\mathbb{C})$ in the same line, the condition is preserved.

Once the loop condition is false, *L1*, *L2*, *L3*, *L4* and the following hold:

$$\forall T' \in \text{trunk}(\mathbb{C})[\text{similarity}(T', T^d) = \text{root}(\mathbb{C})] \quad (\lambda)$$

After the execution of lines 16 and 17 we obtain the postconditions:

T has a profile in \mathbb{C} by *L1*, *L3*, λ and line 17.

\mathbb{C} is atomistic by *L4*.

The similarity condition by *L2*, λ , 16.

Finally, we show that if the *similarity* and *difference* functions terminate, the procedure terminates. At each iteration the selected trunk theory is split into two parts: the similarity S which is added to *Covered* and T'^d whose similarity with T^d is the root theory. Therefore neither of the new trunk theories will be selected again. Since the hierarchy contain only finitely many theories and thus have only a finite number of trunk theories, the loop will eventually terminate. \square

Lemma 17 Let $\mathbb{C} = \langle \mathcal{C}, \leq \rangle$ be an atomistic hierarchy and let T be a theory such that $T \in \mathcal{C}$, T has a profile in \mathbb{C} and $\mathbb{C}' = \langle \mathcal{C} \setminus T, \leq \rangle$ is closed. After executing the *CloseHierarchy*(\mathbb{C}, T) Procedure the resulting hierarchy $\mathbb{C}_{new} = \langle \mathcal{C}_{new}, \leq \rangle$ is closed and atomistic.

Proof: The procedure amounts to finding all theories that have similarities with T , adding the similarities to the hierarchy if they do not already exist and repeating the procedure for all the newly added similarities. It is straightforward to see that if the procedure terminates the hierarchy will be closed. We observe that a discovered similarity is the intersection of the profiles of two existing theories in the hierarchy. Since each theory in the hierarchy is distinct, the newly added theories have profiles that include fewer trunk theories than their parents. Therefore, since the hierarchy has a finite number of trunk theories, at some point no new similarities can be discovered and the procedure terminates. \square

Note that, in general there may be a theory T' in \mathcal{C} such that T is inconsistent with T' . Since T is an extension of the root theory of the hierarchy and the hierarchy is closed, a non-empty similarity between T and T' is guaranteed to exist. In other words, T and T' will at least share the root theory as their similarity, hence the closed property of the hierarchy will be preserved.

Theorem 14 Let $\mathbb{C} = \langle \mathcal{C}, \leq \rangle$ be a closed, atomistic hierarchy and let T be a theory such that $\Sigma(\mathbb{C}) = \Sigma(T)$ and $T \models \text{root}(\mathbb{C})$. If the *UpdateHierarchy*(\mathbb{C}, T) Procedure terminates, the resulting hierarchy $\mathbb{C}_{new} = \langle \mathcal{C}_{new}, \leq \rangle$ is closed, atomistic and T has a profile in \mathbb{C}_{new} .

Proof: Follows directly from Lemmas 16 and 17. \square

Consider the core hierarchy in Figure 5 as example. It consists of two trunk theories $T_{dense_nofinal_partial}$ and $T_{discrete_noinitial_partial}$ that share only the root theory $T_{partial_ordering}$ as their similarity. Now we would like to add a new core theory, $T_{inf_ordering}$, to this hierarchy while maintaining that the core hierarchy is closed and atomistic. In this scenario, the theory we are adding, $T_{inf_ordering}$, is not equivalent to any combination of core theories in the hierarchy. Therefore, *UpdateProfile* is run in order to determine the new subtheory that must be added to the hierarchy to ensure that $T_{inf_ordering}$ is the union of a set of trunk theories. Since $T_{inf_ordering}$ shares similarities with each trunk theory, the procedure breaks down those trunk theories into two subtheories each (one being the similarity shared with $T_{inf_ordering}$ and the other being the difference between the old trunk theory and the similarity). Here, the similarity between $T_{inf_ordering}$ and $T_{dense_nofinal_partial}$ is $T_{nofinal}$, and the difference between $T_{dense_nofinal_partial}$ and the new similarity is $T_{dense_partial}$. For the other trunk theory, we get the new subtheories $T_{noinitial}$ and $T_{discrete_partial}$. It then follows from the definition of trunk theories that these new subtheories are added to the set of trunk theories while the original trunk theories that cover them are no longer considered as trunk theories. Each time a subtheory is added to the hierarchy, it is also added to the profile of the original theory. Here, subtheories $T_{noinitial}$ and $T_{discrete_partial}$ were added to the profile of $T_{discrete_noinitial_partial}$ at the same time they were added as trunk theories in the hierarchy. Finally, the subtheory of $T_{inf_ordering}$ that shares no non-trivial

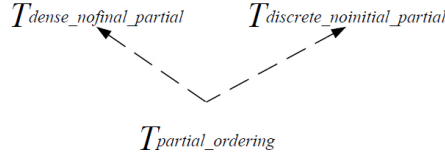


Figure 5. Initial hierarchy with two trunk theories extending the root theory. Dashed lines denote non-conservative extensions.

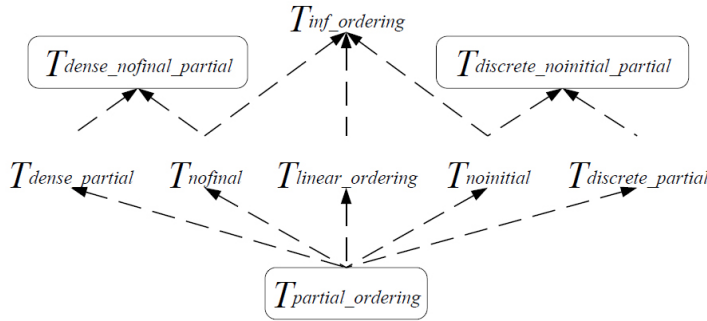


Figure 6. Hierarchy after a new theory $T_{inf_ordering}$ is added using *UpdateHierarchy*. Dashed lines denote non-conservative extensions. Theories that existed in the hierarchy before insertion of the new theory are outlined.

similarity with any trunk theory ($T_{linear_ordering}$) is added to the set of trunk theories. *UpdateProfile* completes and the next step *CloseHierarchy* makes no changes for this example as the hierarchy at this step happens to be closed under similarity. Figure 6 shows the hierarchy after *UpdateHierarchy* is completed, with the original theories before the procedure was run outlined.

8. Towards an Implementation in COLORE

All of the techniques and metatheoretic relationships between theories discussed in this paper are being developed in the context of the COLORE (Common Logic Ontology Repository) project, which is building an open repository of first-order ontologies that serve as a testbed for ontology evaluation and integration techniques, and that can support the design, evaluation, and application of ontologies in first-order logic. All ontologies are specified using Common Logic (ISO 24707), which is a recently standardized logical language for the specification of first-order ontologies and knowledge bases.

At the lowest level are theories of general mathematical structures, such as algebraic structures (e.g. semigroups, groups, rings, vector spaces), and combinatorial structures (e.g. orderings, lattices, graphs). These ontologies serve as the basis for the reducibility of generic ontologies currently within the repository, such as processes, time, mereotopology, and geometry.

The relation $<$ between theories within the same hierarchy and the relation of conservative extensions of theories in different hierarchies are both equivalent to the clo-

sure of `cl-module` within COLORE under the `cl-imports` relation. The notion of reducibility and the \sqsubseteq relation between hierarchies will require extensions to Common Logic.

Recent work in the verification of time ontologies has demonstrated the utility of the procedures introduced in this paper. In [21], the relationships between three different hierarchies of time interval ontologies are explored. New ontologies arose from definable equivalence between hierarchies and similarities within each hierarchy. In [14] and [10], a hierarchy was constructed that combined several ontologies that axiomatize both timepoints and time intervals. One interesting discovery was that the similarity of two ontologies proposed in [16] was actually equivalent to the time ontology independently axiomatized in [25].

9. Summary

This paper has explored the deep connection between ontology modularity and the logical relationships between theories within an ontology repository. Our formal characterization of repository revolves around the notion of hierarchies. Hierarchies are sets of theories that have the same non-logical lexicon; theories within the same hierarchy are related by non-conservative extension. The hierarchies themselves are related by reducibility, by which a theory in a complex hierarchy is definably equivalent to a set of theories in core hierarchies. For each of these relationships, we can define a class of theories that correspond to the intuition of modules. Within hierarchies, the atomic modules are known as trunk theories, which can be considered as a formalization of the ontological design choices for the hierarchy. On the level of repositories, the ordering over hierarchies defines core theories, which are building blocks for more complex ontologies.

Based on these complementary approaches to modularity, we present two sets of semi-automated procedures that support the decomposition of an ontology into modules. The first set of procedures identifies modules as subtheories of the ontology that are definably equivalent to some core theories and that are related to the ontology through reducibility. The second set of procedures takes any core theory and constructs the set of trunk theories whose combination is logically equivalent to the core theory. In addition, this set of trunk theories allows a characterization of which subtheories are shareable and reusable between ontologies.

This work naturally leads to the following challenges:

- Develop techniques for finding new core theories in the case when a reduction of a new theory T does not exist, but T is reducible to an extension of the core hierarchies.
- Incorporate techniques for ontology verification to characterize the models of new theories that are under probation as a result of the *Decomp* Procedure.
- Explore techniques that use the reductions and profiles of theories to generate semantic mappings between ontologies.
- In any hierarchy, the complete set of trunk theories for a core hierarchy corresponds to the axioms of all complete extensions of the root theory. Can these trunk theories in turn be used to design new extensions of existing ontologies that are reducible to the core hierarchies?

10. Acknowledgement

We are grateful to the three reviewers for their detailed comments which helped to improve the clarity of the presentation of the paper.

References

- [1] Burstall, R.M. and Goguen, J.A. (1977) Putting theories together to make specifications. *International Joint Conference on Artificial Intelligence 1977*, pp. 1045–1058.
- [2] Daskalakis, C., Karp, R., Mossel, E., and Riesenfeld, S. (2009) Sorting and Selection in Posets. *Proc. of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 392–401.
- [3] Del Vescovo, C., Parsia, B., Sattler, U., and Schneider, T. (2009) The Modular Structure of an Ontology: An Empirical Study. *Proc. of the Fourth Int. Workshop on Modular Ontologies*, IOS Press.
- [4] Dzhaparidze, G. (1993) A Generalized Notion of Weak Interpretability and the Corresponding Modal Logic. In: *Annals of Pure and Applied Logic* 61:113–160.
- [5] Enderton, H. (1972) *Mathematical Introduction to Logic*, Academic Press.
- [6] Farmer, W. M. (2000) An Infrastructure for Intertheory Reasoning. *Proc. of the Seventeenth Int. Conference on Automated Deduction (CADE-17)*, LNCS 1831, pp. 115–131.
- [7] Farmer, W. M., Guttman, J. D., Thayer, F. J. (1992) Little Theories. *Proc. of the Eleventh Int. Conference on Automated Deduction (CADE-11)*, LNCS 607, pp. 567–581.
- [8] Cuenca Grau, B., Parsia, B., Sirin, E., Kalyanpur, A. Automatic Partitioning of OWL Ontologies Using E-Connections. *Proc. of the Int. Workshop on Description Logics (DL2005)*, 2005.
- [9] Cuenca Grau, B., Parsia, B., Sirin, E. (2009) Ontology Integration Using E-Connections. *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, LNCS 5445, pp. 159–186.
- [10] Grüninger, M. (2011) Verification of the OWL-Time Ontology, in *Proceedings of the Tenth International Semantic Web Conference*. Bonn, Germany.
- [11] Grüninger, M. (2009) Definability and Process Ontologies. *Proc. of the Ninth Int. Symposium on Logical Formalizations of Commonsense Reasoning*, Toronto, Canada.
- [12] Grüninger, M. (2009) Process Ontologies. In: *Handbook of Ontologies, 2nd Edition*, S. Staab (ed.), pp. 419–421, Springer.
- [13] Grüninger, M., Hahmann, T., Hashemi, A., and Ong, D. (2010) Ontology Verification with Repositories. *Proc. of the Sixth Int. Conference on Formal Ontologies in Information Systems (FOIS 2010)*, pp. 317–330, IOS Press.
- [14] Grüninger, M. and Ong, D. (2011) Verification of Time Ontologies with Points and Intervals, in *18th International Symposium on Temporal Representation and Reasoning*, Lübeck, Germany.
- [15] Hashemi, A. (2009) *Using Repositories for Ontology Design and Semantic Mapping*, MASC thesis, University of Toronto, Toronto, Canada.
- [16] Hayes, P. (1996) *A Catalog of Temporal Theories*, Tech Report UIUC-BI-AI-96-01, Univ. Illinois.
- [17] Hodges, W. (1993) *Model Theory*. Cambridge University Press.
- [18] Konev, B., Lutz, C., Walther, D., and Wolter, F. (2009) Formal Properties of Modularisation. *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, LNCS 5445, pp. 25–66.
- [19] Kutz, O. and Mossakowski, T. (2008) Conservativity in structured ontologies. In *Proceedings of the European Conference on Artificial Intelligence 2008*. IOS Press, Amsterdam.
- [20] Mossakowski, T., Maeder, C., Lüttich, K., Wölfl, S. (2004) *The Heterogeneous Tool Set*. Available at www.tzi.de/cofi/hets, University of Bremen.
- [21] Ong, D. and Grüninger, M. (2011) Constructing an Ontology Repository: A Case Study with Theories of Time Intervals, in *Proceedings of the Fifth International Workshop on Modular Ontologies*. Ljubljana.
- [22] Schorlemmer, M. and Kalfoglou, Y. (2008) Institutionalising Ontology-Based Semantic Integration, *Applied Ontology* 3:131–150.
- [23] Sowa, J. (2000) *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Brooks/Cole Publishing Co., Pacific Grove, CA.
- [24] Szczerba, L.W. (1977) Interpretability of Elementary Theories. In: *Logic, Foundations of Mathematics and Computability Theory*, Butts and Hintikka (eds.), pp. 129–145, Springer.
- [25] Vila, L. (2005) Formal Theories of Time and Temporal Incidence, in *Handbook of Temporal Reasoning in Artificial Intelligence*, Fisher, Gabbay, Vila: eds. Elsevier, 2005.