Concept-Based Semantic Difference in Expressive Description Logics

Rafael S. Gonçalves, Bijan Parsia, and Ulrike Sattler

School of Computer Science
University of Manchester
Manchester, United Kingdom

Abstract. Detecting, much less understanding, the difference between two description logic based ontologies is challenging for ontology engineers due, in part, to the possibility of complex, non-local logic effects of axiom changes. First, it is often quite difficult to even determine which concepts have had their meaning altered by a change. Second, once a concept change is pinpointed, the problem of distinguishing whether the concept is directly or indirectly affected by a change has yet to be tackled. To address the first issue, various principled notions of "semantic diff" (based on deductive inseparability) have been proposed in the literature and shown to be computationally practical for the expressively restricted case of $\mathcal{ELH}^r$-terminologies. However, problems arise even for such limited logics as $\mathcal{ALC}$: First, computation gets more difficult, becoming undecidable for logics such as $\mathcal{SROIQ}$ which underly the Web Ontology Language (OWL). Second, the presence of negation and disjunction make the standard semantic difference too sensitive to change: essentially, any logically effectual change always affects all terms in the ontology. In order to tackle these issues, we formulate the central notion of finding the *minimal change set* based on model inseparability, and present a method to differentiate changes which are specific to (thus directly affect) particular concept names. Subsequently we devise a series of computable approximations, and compare the variously approximated change sets over a series of versions of the NCI Thesaurus (NCIt).

1 Introduction

Determining the significant differences between two documents (so-called "diff") is a standard and significant problem across a wide range of activities, notably software development. Standard textual diffing algorithms perform poorly on description logic (DL) based ontologies, both for structural reasons (e.g., ontology serializations, such as those of OWL, tend not to impose stable ordering of axioms), and due to the highly non-local and unintuitive logical effects of changes to axioms. This gave rise to several diff notions for OWL ontologies, encompassing various types of change detection and impact analysis mechanisms. Within change detection there are two key dimensions of change: *syntactic* and *semantic*, leading to syntactic and semantic diffs. The former, e.g. those diffs based on OWL's notion of "structural equivalence" [3,7,11], detect asserted changes

between ontologies, and thus are of interest for, e.g., versioning. In [3] we addressed the problem of identifying and characterising the impact of such asserted changes by, for instance, pinpointing whether each change produces a logical effect. However, that work focused exclusively on axiom level analysis. Since OWL ontologies are sets of axioms, this is a natural level of analysis. However, ontologies often serve as ways to manage controlled vocabularies, that is, the set of axioms is a development time artifact supporting the delivery of a hierarchically organized set of categorical *terms*. In such cases, end users are most directly concerned with changes to the terms themselves and may not even have access to the axioms. Thus, the modeller must not only be aware of the axioms they have touched, but how those changes affect the concepts in the ontology.

For the purpose of determining entailment (and ergo term) differences, recent notions of semantic difference based on conservative extensions have provided a robust theoretical and practical basis for analysing these logical effects [6]. Unfortunately, semantic difference is computationally expensive even for inexpressive logics such as $\mathcal{EL}$. For the very expressive logics such as $\mathcal{SROIQ}$ (the DL underlying OWL 2) it is undecidable [9]. Furthermore, standard semantic difference runs into other difficulties in more expressive logics when we consider differences w.r.t. all terms in both ontologies. In particular, if we compare entailment sets over logics with disjunction and negation we end up with vacuously altered terms: any logically effectual change will alter the meaning of every term.

To address this vacuity problem, we present *a non-trivializable notion of semantic difference of concepts*, which includes *a mechanism for distinguishing directly and indirectly affected concepts*. To address the undecidability of even our refined semantic difference problem for the $\mathcal{SROIQ}$ (i.e., OWL 2) DL, we define *a series of motivated semantic diff approximations for expressive description logics*. These algorithms are evaluated on a select subset of the National Cancer Institute (NCI) Thesaurus (NCIt) corpus, by a comparison of the changes found via the proposed approximations and related approaches. Our experiments show that our strongest approximation, "Grammar diff", finds significantly more changes than all other methods across the corpus, and far more than are identified in the NCIt change logs. We show that distinguishing direct and indirect changes is necessary for making concept based change logs manageable.

2   Preliminaries

We assume the reader to be familiar with ontologies and OWL, as well as the underlying description logics (DLs) [1]. We use *terms* to refer to concept and role names. When comparing two ontologies we refer to them as $\mathcal{O}_1$ and $\mathcal{O}_2$, and their *signatures* (i.e., the set of terms occurring in them) as $\widetilde{\mathcal{O}}_1$ and $\widetilde{\mathcal{O}}_2$, respectively. Throughout this paper we refer to $\widetilde{\mathcal{O}}_1 \cup \widetilde{\mathcal{O}}_2$ as Σ_u. This signature is the natural subject of comparison of terminological changes between two ontologies. The signature of an axiom α is denoted $\widetilde{\alpha}$.

Throughout this paper we use the standard description and first order logic notion of entailment; an axiom α entailed by an ontology $\mathcal{O}$ is denoted $\mathcal{O} \models \alpha$. We

refer to an *effectual* addition (removal) from $\mathcal{O}_1$ to $\mathcal{O}_2$ as an axiom α such that $\alpha \in \mathcal{O}_2$ and $\mathcal{O}_1 \not\models \alpha$ ($\alpha \in \mathcal{O}_1$ and $\mathcal{O}_2 \not\models \alpha$) [3]. Thus two ontologies are logically equivalent, denoted $\mathcal{O}_1 \equiv \mathcal{O}_2$, if there is no effectual change (addition or removal) between $\mathcal{O}_1$ and $\mathcal{O}_2$. The set of *subconcepts* of an ontology $\mathcal{O}$ is recursively defined as all subconcepts found in each axiom of $\mathcal{O}$, plus $\{\top, \bot\}$. W.l.o.g, we define all diff functions asymmetrically, thus to get the full diff between two ontologies we compute $\mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)$ (for additions) and $\mathrm{Diff}(\mathcal{O}_2, \mathcal{O}_1)$ (for removals).

The restriction of an interpretation $\mathcal{I}$ to a set of terms Σ is denoted $\mathcal{I}|_\Sigma$. Two interpretations $\mathcal{I}$ and $\mathcal{J}$ coincide on a signature Σ (denoted $\mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$) if $\Delta^\mathcal{I} = \Delta^\mathcal{J}$ and $t^\mathcal{I} = t^\mathcal{J}$ for each $t \in \Sigma$.

Throughout this paper we use the notion of model conservative extension (mCE) [2,9], and associated inseparability relation [13]. The axioms expressible in a DL $\mathcal{L}$ over a set of terms Σ is denoted $\mathcal{L}(\Sigma)$.

Definition 1. *Given two ontologies $\mathcal{O}_1$, $\mathcal{O}_2$ over a DL $\mathcal{L}$, and a signature Σ.*

(1) $\mathcal{O}_2$ *is model Σ-inseparable from $\mathcal{O}_1$ ($\mathcal{O}_1 \equiv^{mCE}_\Sigma \mathcal{O}_2$) if*
$$\{\mathcal{I}|_\Sigma \mid \mathcal{I} \models \mathcal{O}_1\} = \{\mathcal{J}|_\Sigma \mid \mathcal{J} \models \mathcal{O}_2\}$$

(2) $\mathcal{O}_2$ *is deductive Σ-inseparable from $\mathcal{O}_1$ w.r.t. $\mathcal{L}$ ($\mathcal{O}_1 \equiv^\mathcal{L}_\Sigma \mathcal{O}_2$) if*
$$\{\alpha \in \mathcal{L}(\Sigma) \mid \mathcal{O}_1 \models \alpha\} = \{\alpha \in \mathcal{L}(\Sigma) \mid \mathcal{O}_2 \models \alpha\}$$

(3) $\mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)^\mathcal{L}_\Sigma = \{\eta \in \mathcal{L}(\Sigma) \mid \mathcal{O}_1 \not\models \eta$ *and* $\mathcal{O}_2 \models \eta\}$

Note that $\mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)^\mathcal{L}_\Sigma = \emptyset$ if and only if $\mathcal{O}_1 \equiv^\mathcal{L}_\Sigma \mathcal{O}_2$. Also, bear in mind (esp. for the running example) that $\mathcal{O}_1 \not\equiv^\mathcal{L}_\Sigma \mathcal{O}_2$ implies $\mathcal{O}_1 \not\equiv^{mCE}_\Sigma \mathcal{O}_2$. In the remainder of this paper we use $\mathcal{SROIQ}$ General Concept Inclusions (GCIs) for $\mathcal{L}$, and omit $\mathcal{L}$ if this is clear from the context.

3   Exisiting Semantic Diff

$\mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ alone, if non-empty, tells us that there are new entailments expressed in the designated signature, but does not pick out specific terms in that signature. The CEX [6] diff method focuses on elements of $\mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$, so called *witness axioms*, with specific forms — subsumptions with an atomic left hand (resp. right hand) side, i.e., of the form $A \sqsubseteq C$ (resp. $C \sqsubseteq A$) where A is atomic and C is a possibly complex concept, called the *witness concept*. All terms that appear in those positions in axioms in $\mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ form the set of affected terms (denoted $\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$). By restricting attention to changes to individual terms (rather than to sets of terms together), CEX 1) becomes decidable[1], 2) produces manageable diff reports ($\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ is a subset of the signature, not of the powerset of the signature), and 3) the diff report is nicely interpretable. CEX gets interpretability both by focusing on changes to individual terms in themselves (instead of on coordinated changes to sets of terms) and by exploiting the natural directionality of its witness axioms.

---

[1] At least for the restricted case of acyclic $\mathcal{ELH}^r$ terminologies ($\mathcal{EL}$ extended with role inclusions and range restrictions).

CEX divides $AT(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ into specialised, denoted $AT(\mathcal{O}_1, \mathcal{O}_2)_\Sigma^L$, and generalised, designated $AT(\mathcal{O}_1, \mathcal{O}_2)_\Sigma^R$, concept names depending on whether a term appears on the left or right hand side of a witness axiom (the same term may appear in both). The CEX algorithm is sound and complete: If there is a witness axiom for a term A in the deductive closure of $\mathcal{O}_2$, then CEX will find it.

The computational complexity of deciding Σ-entailment is undecidable for expressive DLs such as $\mathcal{SROIQ}$. For $\mathcal{EL}$ it is already ExpTime-complete [10], while for $\mathcal{ALC}$, $\mathcal{ALCQ}$, and $\mathcal{ALCQI}$ it is 2ExpTime-complete [9]. Thus, CEX is computationally infeasible for expressive logics. Moreover, when considering differences over Σ_u, a direct extension of Σ-difference for more expressive logics such as $\mathcal{ALC}$ would be futile; when we step beyond $\mathcal{EL}$ as a witness language into more expressive logics with disjunction and negation, if $\mathcal{O}_1 \not\equiv \mathcal{O}_2$ then $AT(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ contain all terms in Σ. Consider the following $\mathcal{EL}$ ontologies: $\mathcal{O}_1 = \{A \sqsubseteq B\}$, and $\mathcal{O}_2 = \{A \sqsubseteq B, C \sqsubseteq D\}$. Clearly $\mathcal{O}_2$ is a conservative extension of $\mathcal{O}_1$ w.r.t. $\Sigma = \{A, B\}$, but if we consider Σ_u then that is no longer the case; a witness axiom for the separability would be, e.g., $\eta := A \sqsubseteq \neg C \sqcup D$. This witness "witnesses" a change to every concept $A' \in \Sigma_u$; for each witness axiom $\eta' : A' \sqsubseteq \neg C \sqcup D$ we have that $\mathcal{O}_1 \not\models \eta'$, while $\mathcal{O}_2 \models \eta'$. Such a witness would suffice to pinpoint, according to Σ-difference, that all terms in Σ_u have changed: $AT(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma_u} = \Sigma_u$ since $\top \sqsubseteq \neg C \sqcup D$. Consequently, this kind of witness is uninteresting for any particular concept aside from $\top$. Likewise, a change $A \sqsubseteq \bot$ implies that, for all B in the signature of the ontology in question, we have that $A \sqsubseteq B$. Yet these consequences are of no interest to any concept B.

Similar to the case of the least common subsumer [8], the presence of disjunction (and negation) trivialises definitions that are meaningful in less expressive logics. Thus we need to refine our diff notion when dealing with propositionally closed witness languages.

A simple approach to coping with the dual problems of computational difficulty and triviality is to use a relatively inexpressive witness language. For example, ContentCVS [5] computes an approximation of $AT(\mathcal{O}_1, \mathcal{O}_2)_\Sigma^L$ (i.e., $A \sqsubseteq C$) for OWL 2 DL where C conforms to a specific grammar (B is atomic):

Grammar G_{cvs} : $C \longrightarrow B \mid \exists r.B \mid \forall r.B \mid \neg B$

Grammar G_{cvs} is a bit ad hoc, being based on the designers intuitions of what might be "interesting" but yet inherently finite. In a user study of ContentCVS, users criticised "the excessive amount of information displayed when using larger approximations of the deductive difference" [5]. The users were not presented with the affected terms directly, only via presentation of the witness axioms. ContentCVS avoids triviality by not allowing axiom encoding witness concepts.

4   Semantic Diff

Given the shortcomings of existing methodologies, and the triviality of Σ_u-difference in expressive ontologies, we present a semantic diff method that *a)* determines which concepts have been affected by changes. For exposition reasons, we concentrate on concepts, though roles are easily added. And *b)* identifies

which concepts have been directly (or indirectly) changed. Ideally, a solution to these problems would be *1)* a computationally feasible diff function (for OWL 2 DL), *2)* based on a principled grammar, that *3)* returns those concept names affected by changes between two ontologies, while *4)* distinguishing whether each concept name is directly (or indirectly) specialised and/or generalised.

Consider the toy ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ defined in Table 1; they will be used throughout this section as a running example.

Table 1. Ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$

$\mathcal{O}_1$		$\mathcal{O}_2$	
$\alpha_1:$	$A \sqsubseteq B$	$\beta_1:$	$A \sqsubseteq B$
$\alpha_2:$	$B \sqsubseteq C$	$\beta_2:$	$B \sqsubseteq C \sqcap D$
$\alpha_3:$	$D \sqsubseteq \exists r.E$	$\beta_3:$	$D \sqsubseteq \exists r.E$
$\alpha_4:$	$E \sqsubseteq \forall s.G$	$\beta_4:$	$E \sqsubseteq \forall s.(G \sqcap F)$
$\alpha_5:$	$\exists r.I \sqsubseteq J$	$\beta_5:$	$\exists r.I \sqsubseteq J$
		$\beta_6:$	$\forall t.H \sqsubseteq I$

4.1 Determining the Change Set

Given two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, such that $\mathcal{O}_1 \not\equiv \mathcal{O}_2$ (i.e., there exists at least one effectual change in $\text{Diff}(\mathcal{O}_1, \mathcal{O}_2)$), we know that $\mathcal{O}_1$ and $\mathcal{O}_2$ are not Σ-inseparable (for $\Sigma := \Sigma_u$) w.r.t. model inseparability, i.e., $\mathcal{O}_1 \not\equiv^{mCE}_\Sigma \mathcal{O}_2$ since an effectual change implies some change in semantics. In order to pinpoint this change, we need to find the set of terms Σ' s.t. $\mathcal{O}_1$ is mCE-inseparable from $\mathcal{O}_2$ w.r.t. the remaining signature $\Sigma \setminus \Sigma'$: $\mathcal{O}_1 \equiv^{mCE}_{\Sigma \setminus \Sigma'} \mathcal{O}_2$. Then we know that, from $\mathcal{O}_1$ to $\mathcal{O}_2$, there are no changes in entailments over $\Sigma \setminus \Sigma'$. We refer to this set of terms Σ' as the Minimal Change Set (denoted $\text{MinCS}(\mathcal{O}_1, \mathcal{O}_2)$), in the sense that we can formulate a non-trivial witness axiom η over Σ' s.t. $\mathcal{O}_1 \not\models \eta$ but $\mathcal{O}_2 \models \eta$. Thus we denote these terms as *affected*.

Definition 2 (Minimal Affected Terms). *A set $\Sigma' \subseteq \Sigma$ is a set of minimal affected terms between $\mathcal{O}_1$ and $\mathcal{O}_2$ if:*

$$\mathcal{O}_1 \not\equiv^{mCE}_{\Sigma'} \mathcal{O}_2 \text{ and for all } \Sigma'' \subsetneq \Sigma' : \mathcal{O}_1 \equiv^{mCE}_{\Sigma''} \mathcal{O}_2.$$

The set of all such sets is denoted $\text{MinAT}(\mathcal{O}_1, \mathcal{O}_2)$.

In order to form the minimal change set, we take the union over all sets of affected terms in $\text{MinAT}(\mathcal{O}_1, \mathcal{O}_2)$.

Definition 3 (Minimal Change Set). *The minimal change set, denoted* $\text{MinCS}(\mathcal{O}_1, \mathcal{O}_2)$, *of two ontologies is defined as follows:*

$$\text{MinCS}(\mathcal{O}_1, \mathcal{O}_2) := \bigcup \text{MinAT}(\mathcal{O}_1, \mathcal{O}_2) \, , \text{ and}$$

$$\text{MinCS}(\mathcal{O}_1, \mathcal{O}_2)^C := \{C \mid C \text{ is a concept name in } \text{MinCS}(\mathcal{O}_1, \mathcal{O}_2)\}.$$

From the example ontologies in Table 1 we have that $\{A, D\}$ is a set of minimal affected terms between $\mathcal{O}_1$ and $\mathcal{O}_2$; $\mathcal{O}_1 \not\equiv^{mCE}_{\{A,D\}} \mathcal{O}_2$, because $\mathcal{O}_1 \not\models A \sqsubseteq D$

while $\mathcal{O}_2 \models A \sqsubseteq D$. $\{A, D\}$ is minimal since $\mathcal{O}_1 \equiv_{\{A\}}^{mCE} \mathcal{O}_2$, and similarly for $\{D\}$. Analogous cases can be made for $\{B, D\}$ via witness axiom $B \sqsubseteq D$, $\{E, s, F\}$ via $E \sqsubseteq \forall s.F$, $\{r, t, H, J\}$ via $\exists r.\forall t.H \sqsubseteq J$, and finally $\{t, H, I\}$ via β_6. So the minimal change set (restricted to concept names) between these two ontologies is $\mathrm{MinCS}(\mathcal{O}_1, \mathcal{O}_2)^C := \{A, B, D, E, F, H, I, J\}$.

4.2 Characterising Concept Impact

Prior to determining how a concept in a signature Σ has changed (e.g., it has a new superconcept), we employ a diff function Φ which, given two ontologies and Σ, formulates *a set of witness axioms over* Σ, denoted $\Phi \mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$, such that, for each $\eta \in \Phi \mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$: $\mathcal{O}_1 \not\models \eta$ and $\mathcal{O}_2 \models \eta$. Now given such a set $\Phi \mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$, we can tell apart specialised and generalised concepts depending on whether the witness concept is on the right or left hand side of the witness axiom, accordingly. Furthermore, we regard a concept name A as *directly specialised (generalised)* via some witness C if there is no concept name B that is a superconcept (subconcept) of A, and C is also a witness for a change in B. Otherwise A *changed indirectly.*

Definition 4. *A diff function* Φ *returns a subset* $\Phi \mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ *of* $\mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$. *For a diff function* Φ, *the sets of* affected concept names *for a signature* Σ *are:*

$$\Phi\text{-}\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma^\top = \begin{cases} \{\top\} & \textit{if there is a } \top \sqsubseteq C \in \Phi\mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma \\ \emptyset & \textit{otherwise} \end{cases}$$

$$\Phi\text{-}\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma^\bot = \begin{cases} \{\bot\} & \textit{if there is a } C \sqsubseteq \bot \in \Phi\mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma \\ \emptyset & \textit{otherwise} \end{cases}$$

$$\Phi\text{-}\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma^\mathrm{L} = \{A \in \Sigma \mid \textit{there exists } A \sqsubseteq C \in \Phi\mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma \textit{ and} \\ \top \sqsubseteq C \notin \Phi\mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma\}$$

$$\Phi\,\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma^\mathrm{R} = \{A \in \Sigma \mid \textit{there exists } C \sqsubseteq A \in \Phi\mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma \textit{ and} \\ C \sqsubseteq \top \notin \Phi\mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma\}$$

$$\Phi\text{-}\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma = \bigcup\nolimits_{Y \in \{L, R, \top, \bot\}} \Phi\text{-}\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma^Y$$

Given a concept name $A \in \Phi\text{-}\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma^\mathrm{L}$ *(analogously* $A \in \Phi\text{-}\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma^\mathrm{R}$*), and a set of terms* $\Sigma^+ := \Sigma \cup \{\top, \bot\}$, *we define the following notions:*

A direct change *of* A *is a witness* C *s.t.* $A \sqsubseteq C$ $(C \sqsubseteq A) \in \Phi\mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)$ *and there is no* $B \in \Sigma^+$ *s.t.* $\mathcal{O}_2 \models A \sqsubseteq B$ $(\mathcal{O}_2 \models B \sqsubseteq A)$, $\mathcal{O}_2 \not\models A \equiv B$, *and* $B \sqsubseteq C$ $(C \sqsubseteq B) \in \Phi\mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)$.

An indirect change *of* A *is a witness* C *s.t.* $A \sqsubseteq C$ $(C \sqsubseteq A) \in \Phi\mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)$ *and there is at least one* $B \in \Sigma^+$ *s.t.* $\mathcal{O}_2 \models A \sqsubseteq B$ $(\mathcal{O}_2 \models B \sqsubseteq A)$, $\mathcal{O}_2 \not\models A \equiv B$ *and* $B \sqsubseteq C$ $(C \sqsubseteq B) \in \Phi\mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)$.

Concept A *is* purely directly changed *if it is only directly changed (analogously for* purely indirectly changed*).*

As a consequence of Definition 4, $\Phi\text{-AT}(\mathcal{O}_1, \mathcal{O}_2) \subseteq \text{MinCS}(\mathcal{O}_1, \mathcal{O}_2)$. Once again, take as an example the ontologies in Table 1; we have that B is purely directly specialised via witness D: $\mathcal{O}_1 \not\models B \sqsubseteq D$ and $\mathcal{O}_2 \models B \sqsubseteq D$, while A is indirectly specialised via the same witness, since $\mathcal{O}_1 \not\models A \sqsubseteq D$, $\mathcal{O}_2 \models A \sqsubseteq D$, $\mathcal{O}_2 \models A \sqsubseteq B$ and $B \sqsubseteq D \in \text{Diff}(\mathcal{O}_1, \mathcal{O}_2)$. In other words, concept A changes via B. Additionally, the concept D is directly generalised via B, but indirectly generalised via A. Thus D is not purely directly changed, but rather we have a mixed effect on the concept.

The distinction between directly and indirectly affected concept names, in addition to the separation of concepts affected via $\top$ and $\bot$, allows us to overcome the problem described in Section 3, w.r.t. propositionally closed description logics. If there exists a global change to $\top$ (analogously to $\bot$), it is singled out from the remaining localised changes, and its effect is appropriately marked as an indirect change to every concept name. Thus the diff results are no longer "polluted" by vacuous witnesses such as those exemplified and discussed in Section 3. The notion of "change effect" as per Definition 4 is applicable to any diff function Φ that produces a set of witness axioms $\Phi \text{Diff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$.

4.3 Diff Functions

Deciding the minimal change set between two ontologies involves deciding whether, for a given signature Σ, two ontologies are mCE-inseparable w.r.t. Σ. Since mCE-inseparability is undecidable for $\mathcal{SROIQ}$ [9],[2] we devise several sound but incomplete approximations to the problem of computing the minimal change set: to start with, "Subconcept" diff, denoted $\text{SubDiff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$, and "Grammar" diff, denoted $\text{GrDiff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$. The set of differences that would be captured by a simple comparison of concept hierarchies between two ontologies, i.e., differences in atomic subsumptions, is denoted $\text{AtDiff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$. Hereafter we refer to the semantic diff notion used within ContentCVS as $\text{CvsDiff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$.

The $\text{SubDiff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ approximation is based on subconcepts explicitly asserted in the input ontologies, and returns those differences in entailments of type $C \sqsubseteq D$, where C and D are possibly complex concepts from the set of Σ-subconcepts of $\mathcal{O}_1$ and $\mathcal{O}_2$ (see Definition 5). It is at least conceivable that many entailments will involve subconcepts, and, if that is the case, those would be witnesses that the user could understand and, indeed, may have desired. Furthermore, this notion may find entailment differences that would not show up if we restrict ourselves to either atomic subsumptions, or specific forms of entailments (in the manner of $\text{CvsDiff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$). The restriction to forms of concepts explicit in either ontology, however, limits the amount of change captured. In our ontologies in Table 1, e.g., the change to concept D: $\mathcal{O}_1 \not\models D \sqsubseteq \exists r.\forall s.F$, while $\mathcal{O}_2 \models D \sqsubseteq \exists r.\forall s.F$, cannot be captured by $\text{SubDiff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$. However, the rationale behind this approach is that we could detect other kinds of change in a principled and relatively cheap way, e.g., we have that $\mathcal{O}_1 \not\models A \sqsubseteq \exists r.E$, and $\mathcal{O}_2 \models A \sqsubseteq \exists r.E$. Obviously we could arbitrarily extend our entailment grammar to, for instance, a subset of the $\mathcal{SROIQ}$ closure, thus finding even

---

[2] Indeed mCE-inseparability is already undecidable for general $\mathcal{EL}$ ontologies [10].

more witnesses. Though our aim is to capture as much change as possible while maintaining both computational feasibility and legibility of witness axioms.

Nevertheless, in order to avoid only considering witnesses in their explicitly asserted form, we extend the previous diff notion to $\mathrm{GrDiff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$, which detects differences in additional types of entailments using the following grammars (where SC, SC' stand for subconcepts of $\mathcal{O}_1 \cup \mathcal{O}_2$, and r a role name):

Grammar G_L : $C \longrightarrow SC \mid SC \sqcup SC' \mid \exists r.SC \mid \forall r.SC \mid \neg SC$
Grammar G_R : $C \longrightarrow SC \mid SC \sqcap SC' \mid \exists r.SC \mid \forall r.SC \mid \neg SC$

$\mathrm{GrDiff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ combines the basic intuitions about interesting logical forms with the ontology specific information available from $\mathrm{SubDiff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ to be somewhat less ad hoc. By restricting fillers of the restrictions to the (inherently) finite set of subconcepts, we ensure termination. The grammars are slightly optimized to avoid pointless redundancies, such as testing for $A \sqsubseteq C \sqcap D$ which is equivalent to $A \sqsubseteq C$ and $A \sqsubseteq D$. It is not obvious how to reasonably extend these grammars to incorporate features such as number restrictions.

In terms of computational complexity, there are two dimensions to be considered: 1) the complexity of deciding entailment in the input language, and 2) the number of entailment tests. Regarding the latter, the maximum number of *candidate witness axioms* is polynomial in the number of the inputs' subconcepts, namely quadratic for $\mathrm{SubDiff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ and cubic for $\mathrm{GrDiff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$.

The semantic difference between ontologies w.r.t. each mentioned diff function, including CEX and $\mathrm{CvsDiff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$, is boiled down to finding an entailment that holds in $\mathcal{O}_2$ but not $\mathcal{O}_1$; what varies between each function is the kind of entailment grammar used, which in turn dictates the computational feasibility of the diff function.

Definition 5. *Given two ontologies, a diff function Φ, and a signature Σ, the set of Σ-differences is:*

$$\Phi\,\mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma := \{\eta \in \Phi\text{-}ax \mid \mathcal{O}_1 \not\models \eta \wedge \mathcal{O}_2 \models \eta \wedge \widetilde{\eta} \subseteq \Sigma\}$$

where the set Φ-ax is defined as follows:

$$
\begin{aligned}
&\textit{if } \Phi = At, &&\{C \sqsubseteq D \mid C, D \in \Sigma\} \\
&\textit{if } \Phi = Sub, &&\{C \sqsubseteq D \mid C, D \textit{ subconcepts in } \mathcal{O}_1 \cup \mathcal{O}_2\} \\
&\textit{if } \Phi = Gr, &&\{C \sqsubseteq D \mid D \textit{ a concept over } G_L, \textit{ or } C \textit{ a concept over } G_R\} \\
&\textit{if } \Phi = Cvs, &&\{C \sqsubseteq D \mid C \in \Sigma \textit{ and } D \textit{ a concept over } G_{cvs}\} \\
&\textit{if } \Phi = CEX, &&\{C \sqsubseteq D \mid C, D \textit{ subconcepts in } \mathcal{L}(\Sigma)\}
\end{aligned}
$$

Applying the diff functions At, Sub, and Gr from Definition 5 to our example ontologies from Table 1, we get the sets of affected terms described in Table 2.

The differences in atomic subsumptions are easily identifiable, and follow from axioms α_1, α_2 in $\mathcal{O}_1$ and their β_1, β_2 counterparts in $\mathcal{O}_2$. In addition to these, $\mathrm{SubDiff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ pinpoints the axioms β_4 and β_5 as new entailments in $\mathcal{O}_2$, thus concept E is regarded as specialised via β_4, and I generalised via β_5. Finally

Table 2. Affected concepts (specialised, generalised and total) between $\mathcal{O}_1$ and $\mathcal{O}_2$ according to the mentioned diff notions

	$\Phi = At$	$\Phi = Sub$	$\Phi = Gr$	$\text{MinCS}(\mathcal{O}_1, \mathcal{O}_2)^C$
$\Phi\text{-AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma^L$	$\{A, B\}$	$\{A, B, E\}$	$\{A, B, D, E\}$	-
$\Phi\text{-AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma^R$	$\{D\}$	$\{D, I\}$	$\{D, I, J\}$	-
$\Phi\text{-AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$	$\{A, B, D\}$	$\{A, B, D, E, I\}$	$\{A, B, D, E, I, J\}$	$\Sigma \setminus \{C, G\}$

$\text{GrDiff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ spots two more affected concepts: D is specialised via witness axiom $D \sqsubseteq \exists r.\forall s.(F \sqcap G)$, and J is generalised via $\exists r.\forall t.H \sqsubseteq J$. Taking into account $\text{MinCS}(\mathcal{O}_1, \mathcal{O}_2)^C$, it is evident that the more we expand our entailment grammar, the closer we get to the actual change set, while remaining decidable – as long as the language generated by the grammar is finite. We already discussed the computational upper bound above, and will comment on the performance of our implementation in Section 5.

It is not hard to see that there are subset relations between each diff, and the set $\text{MinCS}(\mathcal{O}_1, \mathcal{O}_2)$ that they approximate, as per Lemma 1:

Lemma 1. *Given two ontologies and a signature Σ:*

$$\text{At-AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma \subseteq \text{Sub-AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma \subseteq \text{Gr-AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma \subseteq \text{MinCS}(\mathcal{O}_1, \mathcal{O}_2)$$
$$\text{Cvs-AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma \subseteq \text{Gr-AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma \subseteq \text{MinCS}(\mathcal{O}_1, \mathcal{O}_2)$$

As for CEX, its current implementation only takes as input acyclic $\mathcal{ELH}^r$ *terminologies*, that is, $\mathcal{ELH}^r$ TBoxes which are 1) acyclic and 2) every concept appears (alone) on the left-hand side of an axiom exactly once. In order to apply CEX to knowledge bases that are more expressive than $\mathcal{ELH}^r$ terminologies, one must rely on approximation algorithms. An $\mathcal{EL}$ approximation does not suffice, as there may exist cycles, GCIs, or more than one axiom with the same left hand side. Therefore, as a means to apply CEX to expressive ontologies, we use two $\mathcal{ELH}^r$ approximations.

Definition 6. *For an ontology $\mathcal{O}$, we define the approximation function $\mathcal{ELH}^r\text{App}_1(\mathcal{O})$ that approximates $\mathcal{O}$ into $\mathcal{ELH}^r$ as follows:*

(a) Remove all axioms with a non-atomic left hand side and all non-$\mathcal{EL}$ axioms.
(b) If there is an equivalence axiom with an atomic left or right hand side X, and a non-empty set of subsumptions Ψ that have X on their left hand side, remove all axioms in Ψ.
(c) Break cycles by non-deterministically removing axioms in cycles until the resulting ontology is cycle-free.
(d) Remove all but one axiom with a given atomic left-hand side.

The approximation function $\mathcal{ELH}^r\text{App}_2(\mathcal{O})$ is the same as $\mathcal{ELH}^r\text{App}_1(\mathcal{O})$ but with Step (d) *replaced with* (d') *as follows:*

*(d') Replace the set of axioms with a common left hand side concept A, e.g.,
$\{A \sqsubseteq C, A \sqsubseteq D\}$, with a subsumption between A and the conjunction of all
concepts on the right hand side of all such axioms, e.g., $A \sqsubseteq C \sqcap D$.*

Based on these approximation algorithms, we can now use CEX as a sub-routine
in a diff function for non-$\mathcal{ELH}^r$ ontologies.

Definition 7. *Given two ontologies, a signature Σ, and an $\mathcal{ELH}^r$ approxima-
tion function $\mathcal{ELH}^r\mathrm{App}_i(\mathcal{O})$, the set of Σ-differences $\mathrm{Cex}_i\mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ is:*

1. *For each $j \in \{1,2\}$, execute $\mathcal{ELH}^r\mathrm{App}_i(\mathcal{O}_j)$, resulting in $\mathcal{O}'_j$.*
2. *Apply CEX to $(\mathcal{O}'_1, \mathcal{O}'_2, \Sigma)$, resulting in the change set: TempCS.*
3. *For each $\alpha \in TempCS$, add α to $\mathrm{Cex}_i\mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ if $\mathcal{O}_1 \not\models \alpha$ and $\mathcal{O}_2 \models \alpha$.*

Given the loss of axioms during the input approximation step (via the $\mathcal{ELH}^r$
approximation functions), especially due to its non-deterministic nature, we may
well introduce spurious changes. Thus Step 3 in Definition 7 is designed to ensure
that changes detected within the $\mathcal{ELH}^r$ approximations (obtained in Step 2) are
sound changes w.r.t. the whole (untouched) input ontologies. In other words,
to verify which detected changes are due to the approximation step. Obviously,
this approximation-based procedure throws away a lot of information and is not
deterministic. However, even such an approximation can offer useful insight, par-
ticularly if it finds changes that other methods do not. There are more elaborate
existing approximation approaches (e.g., [12]), but they generally do not pro-
duce $\mathcal{ELH}^r$ terminology, so their use requires either changing the approximation
output or updating CEX to take non-terminological $\mathcal{EL}$ input.

5   Empirical Results

The object of our evaluation is a subset of the NCIt corpus used in [3], with
expressivity ranging from $\mathcal{ALCH}(\mathcal{D})$ to $\mathcal{SH}(\mathcal{D})$. More specifically, we take into
account 14 versions (out of 103 in the whole corpus) of the NCIt (from release
05.06f to 06.08d), and perform consecutive, pairwise comparisons between those
versions which contain concept-based change logs. These versions range from
$\approx$70,000 to $\approx$85,000 logical axioms, and from $\approx$43,000 to $\approx$57,000 concept names.
In order to investigate the applicability of our approach we *(1)* compare the re-
sults obtained via our approximations with those output by $\mathrm{Cex}_1\mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$,
$\mathrm{Cex}_2\mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ and $\mathrm{CvsDiff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$, *(2)* compare the number of (purely)
directly and indirectly affected concepts, and, finally, *(3)* inspect whether the
devised approximations capture changes not reported in the NCIt change logs.

The experiment machine used is an Intel Xeon Quad-Core 3.20GHz with 16Gb
DDR3 RAM. The system runs Mac OS X 10.6.8, Java Virtual Machine (JVM
v1.5), and all tests were run using the OWL API (v3.2.4) [4].

Table 3. Number of concepts processed per minute by each diff function Φ

	$\Phi = \mathrm{Cex}_1$	$\Phi = \mathrm{Cex}_2$	$\Phi = At$	$\Phi = Sub$	$\Phi = Cvs$	$\Phi = Gr$
#Concepts/Min.	151	143	13,547	127	58	50

In terms of computation times, the average number of concepts processed per minute by each diff function is shown in Table 3.[3] The typical total time ranges from seconds for $\mathrm{AtDiff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$, to about 30 minutes for the $\mathrm{Cex}_1\mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$, $\mathrm{Cex}_2\mathrm{Diff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$, and $\mathrm{Sub}\text{-}\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ to hours for $\mathrm{GrDiff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ and $\mathrm{CvsDiff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$.

$\mathrm{GrDiff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ and $\mathrm{CvsDiff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ are rather computationally expensive; the current implementation uses a naive "generate-and-test" approach, where, for each concept, we generate candidate witnesses from the grammar until a witness is found or we exhaust the set. There is clearly considerable scope for optimization.
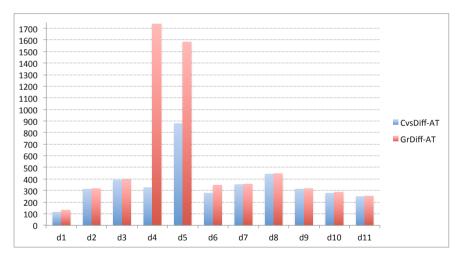
5.1 Diff Comparison

The comparison of each diff w.r.t. the total number of affected concept names found is presented in Table 4. Figure 1 shows a comparison of the number of affected concept names found by $\mathrm{Cvs}\text{-}\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ and $\mathrm{Gr}\text{-}\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ within the randomly selected signatures. Due to computational issues regarding $\mathrm{GrDiff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ and $\mathrm{CvsDiff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$, instead of comparing each pair of NCIt versions w.r.t. Σ_u, we take a random sample of the terms in Σ_u (generally $n \approx 1800$) such that a straightforward extrapolation allows us to determine that the true proportion of changed terms in Σ_u lies in the confidence interval (+-3%) with a 99% confidence level.

In general, $\mathrm{Gr}\text{-}\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$, even taking into account the confidence interval, consistently detects more affected concepts (both L and R, i.e., specialised and generalised, accordingly) than all other diffs. The CEX-based approximation $\mathrm{Cex}_1\text{-}\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ performs poorly across the board, consistently capturing less affected concepts than even a comparison of atomic subsumptions. The second CEX-based approximation $\mathrm{Cex}_2\text{-}\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$, however, typically detects more affected terms than $\mathrm{At}\text{-}\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$, apart from two cases ($d_6$ and d_{10}), but still less than $\mathrm{Sub}\text{-}\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$. Regardless of this result, it is not the case that $\mathrm{Sub}\text{-}\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ is always better than $\mathrm{Cex}_2\text{-}\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$, as the latter actually detects more generalised concept names than $\mathrm{Sub}\text{-}\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ in all but one case. The gathered evidence suggests that indeed combining these approaches would perhaps result in a preferable semantic diff solution than one or the other, as exhibited by the higher average coverage of 59% in $\mathrm{Un}\text{-}\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ (although only an 8% increase w.r.t. to the average coverage of $\mathrm{Sub}\text{-}\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$). As expected, $\mathrm{Gr}\text{-}\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ captures more specialised

---

[3] Note that, originally, $\mathrm{CvsDiff}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ only computes $\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma^L$, but in order to provide a direct comparison with the diffs here proposed we also compute $\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)_\Sigma^R$ according to the G_{cvs} grammar.

Table 4. Number of affected concept names, $AT(\mathcal{O}_i, \mathcal{O}_{i+1})_\Sigma$, found by each diff function (in addition to Un-AT := $\{Cex_1\text{-}AT \cup Cex_2\text{-}AT \cup Sub\text{-}AT\}$) w.r.t. $\Sigma := \Sigma_u$, and their respective coverage w.r.t. $Gr\text{-}AT(\mathcal{O}_i, \mathcal{O}_{i+1})_\Sigma$. At this point, no distinction is made between direct and indirect changes.

Comparison	Cex_1-AT	Cex_2-AT	At-AT	Sub-AT	Un-AT	Gr-AT
d_1	1,134	1,922	1,416	2,131	3,311	43,096
d_2	877	1,746	1,208	1,816	3,307	43,928
d_3	5,415	6,287	6,135	6,528	8,818	45,639
d_4	2,145	6,198	3,676	45,932	45,932	46,929
d_5	3,964	7,656	4,978	15,691	15,758	48,075
d_6	2,298	3,718	3,923	6,203	8,570	48,629
d_7	1,893	3,393	3,217	6,330	7,508	49,189
d_8	6,387	7,397	6,806	7,428	8,957	54,870
d_9	1,655	4,460	2,745	5,329	6,913	55,555
d_{10}	1,512	3,681	4,553	6,415	8,147	55,948
d_{11}	1,102	3,026	1,714	4,325	5,916	57,036
Avg. Cov.	18%	23%	27%	55%	59%	
Min. Cov.	3%	8%	5%	18%	21%	
Max. Cov.	47%	49%	52%	100%	100%	

Fig. 1. Comparison of number of specialised concepts found by $Cvs\text{-}AT(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ and $Gr\text{-}AT(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ within the signature samples of the NCIt (y-axis: number of concept names, x-axis: comparison identifier)
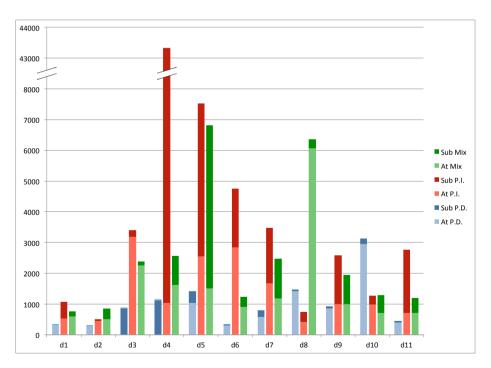
concepts than $Cvs\text{-}AT(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ in all cases, evidenced in Figure 1.[4] Both of these diff approaches resolve all terms in the random signature as generalised.

---

[4] Note that, since $Gr\text{-}AT(\mathcal{O}_1, \mathcal{O}_2)_\Sigma^R = Cvs\text{-}AT(\mathcal{O}_1, \mathcal{O}_2)_\Sigma^R$, we only present in Figure 1 the results of $Gr\text{-}AT(\mathcal{O}_1, \mathcal{O}_2)_\Sigma^L$ and $Cvs\text{-}AT(\mathcal{O}_1, \mathcal{O}_2)_\Sigma^L$.

Thus the projected value implies that nearly, if not every term in the full signature has been generalised.

5.2   Splitting Direct and Indirect Changes

Having the results of each diff at hand, i.e., the set of affected concepts and, for each of these, the set of witnesses, we can then tell apart those concept names that are directly, indirectly, or both directly and indirectly affected. Note that, as an optimisation within the implementation of $\mathrm{GrDiff}(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma}$ and $\mathrm{CvsDiff}(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma}$ diff, we only compute one witness per concept. Thus we do not possess the full set of witnesses, making the distinction of directly and indirectly affected concepts possibly unsound and incomplete. As such, we apply this distinction only to $\mathrm{SubDiff}(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma}$ and $\mathrm{AtDiff}(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma}$. Figure 2 shows the total number of purely direct, purely indirect, and both directly and indirectly affected concepts found within $\mathrm{At\text{-}AT}(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma}$ and $\mathrm{Sub\text{-}AT}(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma}$. Note that the size of $\mathrm{SubDiff}(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma}$ can be smaller than $\mathrm{AtDiff}(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma}$, as in versions v3 and v4. For these particular cases, we bring to the front the smaller value (i.e. $\mathrm{SubDiff}(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma}$), and the value of $\mathrm{AtDiff}(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma}$ becomes

Fig. 2. Comparison of purely directly ("P.D."), purely indirectly ("P.I."), and both directly and indirectly (denoted "Mix") affected concepts found within $\mathrm{At\text{-}AT}(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma}$ (denoted "At"), and $\mathrm{Sub\text{-}AT}(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma}$ (denoted "Sub") in NCIt versions (y-axis: number of concept names, x-axis: comparison identifier)

the increment. Also, this figure presents the total number of changes; the union of $\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)^{\mathrm{L}}_{\Sigma}$ and $\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)^{\mathrm{R}}_{\Sigma}$.

In general, the number of purely directly changed concepts is much smaller than the number of purely indirect or mixed. One case is particularly surprising: Sub-AT$(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma}$ contains 43,326 purely indirect changes in v4, and only 1,122 purely direct ones. In an ontology engineering scenario, where one or more people are required to analyse such change sets, having this mechanism for isolating changes of most interest is conceivably a preferable means to analyse a change set, in addition to providing a basis for producing more intelligible change logs with impact analysis.

5.3   Analysis of the NCIt Change Logs

The change logs supplied with each version of the NCIt contain those concept names which were subject to changes. However, it is unclear whether each reported change may also (or solely) relate to annotation changes. It could be the case that a reported concept change is purely ineffectual as well. In spite of this ambiguity, it should be expected that a change log contains at least those concept names that are directly changed, and this is what we aim to find out in our next experiment; we extract the concept names mentioned in the change log, and verify whether the obtained direct changes for each NCIt comparison are contained in said change logs. The results are shown in Table 5, comparing the number of directly affected concept names found within At-AT$(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma}$ and Sub-AT$(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma}$, and how many of those are not present n the NCIt change logs. Overall, we determined that the change logs are missing a lot of direct

Table 5. Number of directly affected concepts *1)* in $\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)^{\mathrm{L}}_{\Sigma}$ (denoted "L"), *2)* in $\mathrm{AT}(\mathcal{O}_1, \mathcal{O}_2)^{\mathrm{R}}_{\Sigma}$ (denoted "R"), *3)* in the union of those two sets (denoted "Total"), and *4)* that do not appear in the NCIt change logs (denoted "Missed in Log"), found by AtDiff$(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma}$ and SubDiff$(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma}$ for $\Sigma := \Sigma_u$

NCIt version	At-AT$(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma}$				Sub-AT$(\mathcal{O}_1, \mathcal{O}_2)_{\Sigma}$			
	L	R	Total	Missed in Log	L	R	Total	Missed in Log
d_1	646	294	896	798	820	298	1,060	953
d_2	565	274	772	149	1,147	294	1,298	211
d_3	2,321	891	2,991	315	2,791	898	3,090	445
d_4	1,624	1,187	2,683	190	2,725	1,198	2,814	432
d_5	1,555	1,009	2,465	243	8,038	1,186	9,142	317
d_6	890	385	1,130	199	1,306	401	1,485	199
d_7	1,190	704	1,637	273	2,720	780	2,935	511
d_8	6,075	1,421	6,389	5,546	6,411	1,465	6,693	5,723
d_9	1,481	420	1,766	207	2,607	478	2,782	322
d_{10}	3,321	370	3,579	216	4,964	427	5,217	298
d_{11}	753	378	1,043	300	1,404	472	1,643	582
Total	20421	7,333	25,351	8,436	34,933	7,897	38,159	9,993

changes. More specifically, on average, At-AT$(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ contains 767 directly affected concept names not mentioned in the change logs, while Sub-AT$(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ uncovers 908 such concept names per NCIt comparison.

Subsequently we verify whether the affected concepts in Cex$_1$-AT$(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$, Cex$_2$-AT$(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$, At-AT$(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ and Sub-AT$(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ are contained in the NCIt change logs. This is presented in Table 6. Overall we see that none of the diffs captures the exact number of reported concept changes in the logs. The maximum coverage of the change log occurs in comparisons d_4 and d_5, where Sub-AT$(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ captures 96% and 91% of the concept names mentioned in the logs, accordingly. By taking the union of affected concepts found by the CEX-based approximations and Sub-AT$(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$, the average coverage of the change logs increases to 73%.

Table 6. Number of affected concept names, AT$(\mathcal{O}_i, \mathcal{O}_{i+1})_\Sigma$, found by each diff function (in addition to Un-AT := {Cex$_1$-AT$\cup$Cex$_2$-AT$\cup$Sub-AT}) w.r.t. $\Sigma := \Sigma_u$ within the NCIt change logs

NCIt Version	Change Log	Cex$_1$-AT	Cex$_2$-AT	At-AT	Sub-AT	Un-AT
d_1	2,159	107	168	103	126	269
d_2	1,399	520	773	725	974	1,013
d_3	4,234	2,497	2,973	3,102	3,148	3,150
d_4	8,447	1,327	1,598	2,734	8,117	8,117
d_5	3,847	1,595	2,655	2,602	3,503	3,504
d_6	2,470	866	1,147	1,141	1,312	1,406
d_7	5,302	1,217	1,253	1,982	2,668	2,699
d_8	2,556	688	885	875	993	1,003
d_9	3,945	1,060	2,205	1,878	2,530	2,755
d_{10}	6,046	978	3,824	3,551	4,076	6,046
d_{11}	2,065	628	764	853	1,091	1,168
Avg. Coverage		27%	43%	46%	67%	73%

6 Discussion

First thing to notice is that Sub-AT$(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ finds more affected concepts than At-AT$(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$, Cex$_1$-AT$(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$, and Cex$_2$-AT$(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$, while often not reaching close to the projected values of Gr-AT$(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ (the average coverage being 55%). The latter captures more specialised concepts within the selected signatures than Cvs-AT$(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$, while the number of generalised concepts is the same for both diffs (i.e., the full signature).

Considering the high number of affected concepts in Sub-AT$(\mathcal{O}_1, \mathcal{O}_2)_\Sigma$ on comparisons d_4 and d_5 of the NCIt, one can argue that analysing such a change set would be difficult. By categorising concept names in the change set according to whether they are directly or indirectly affected, we get a succinct representation of a change set, thus significantly reducing information overload. Note that,

e.g., in d_4 there are 45,825 specialised concepts, out of which there are only 78 purely directly specialised concepts, and the majority of the remainder are purely indirectly specialised concepts (43,100). Similarly in d_5, from 15,254 specialised concepts there are only 1,527 purely direct specialisations. Immediately we see that this mechanism can provide an especially helpful means to 1) assist change analysis, by, e.g., confining the changes shown upfront to only (purely) direct ones, and 2) generate more informative concept-based change logs.

7   Conclusions

We have formulated the problem of finding the set of affected terms between ontologies via model inseparability, and presented feasible approximations to finding this set. We have shown that each of the approximations can find considerably more changes than those visible in a comparison of concept hierarchies. Both sound approximations devised capture more changes than the CEX-based approximations. The restrictions imposed by CEX on the input ontologies make change-preserving approximations a challenge, as we have seen in our attempt to reduce the NCIt to $\mathcal{EL}$ in a less naive way.

The proposed distinction between (purely) direct and indirect changes allows users to focus on those changes which are specific to a given concept, in addition to masking possibly uninteresting changes to any and all concept names (such as those obtained via witnesses constructed with negation and disjunction), thereby making change analysis more straightforward. As demonstrated by the NCIt change log analysis, we have found a (often high) number of direct changes that are not contained in the NCIt change logs, which leads us to believe the recording of changes does not seem to follow from even a basic concept hierarchy comparison, but rather a seemingly ad hoc mechanism.

In future work we aim to optimise the devised approximations so as to compare all NCIt versions w.r.t. their signature union, and combine the information from this concept oriented diff with our axiom oriented one.

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
2. Ghilardi, S., Lutz, C., Wolter, F.: Did I damage my ontology? A case for conservative extensions in description logics. In: Proc. of KR 2006 (2006)
3. Gonçalves, R.S., Parsia, B., Sattler, U.: Categorising logical differences between OWL ontologies. In: Proc. of CIKM 2011 (2011)
4. Horridge, M., Bechhofer, S.: The OWL API: A Java API for working with OWL 2 ontologies. In: Proc. of OWLED 2009 (2009)
5. Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I., Berlanga Llavori, R.: Supporting concurrent ontology development: Framework, algorithms and tool. Data and Knowledge Engineering 70(1), 146–164 (2011)

6. Konev, B., Walther, D., Wolter, F.: The Logical Difference Problem for Description Logic Terminologies. In: Armando, A., Baumgartner, P., Dowek, G. (eds.) IJCAR 2008. LNCS (LNAI), vol. 5195, pp. 259–274. Springer, Heidelberg (2008)
7. Křemen, P., Šmíd, M., Kouba, Z.: OWLDiff: A practical tool for comparison and merge of OWL ontologies. In: Proc. of DEXA 2012 (2011)
8. Küsters, R.: Non-Standard Inferences in Description Logics. LNCS (LNAI), vol. 2100. Springer, Heidelberg (2001)
9. Lutz, C., Walther, D., Wolter, F.: Conservative extensions in expressive description logics. In: Proc. of IJCAI 2007 (2007)
10. Lutz, C., Wolterinst, F.: Conservative Extensions in the Lightweight Description Logic $\mathcal{EL}$. In: Pfenning, F. (ed.) CADE 2007. LNCS (LNAI), vol. 4603, pp. 84–99. Springer, Heidelberg (2007)
11. Malone, J., Holloway, E., Adamusiak, T., Kapushesky, M., Zheng, J., Kolesnikov, N., Zhukova, A., Brazma, A., Parkinson, H.E.: Modeling sample variables with an experimental factor ontology. Bioinformatics 26(8), 1112–1118 (2010)
12. Ren, Y., Pan, J.Z., Zhao, Y.: Soundness Preserving Approximation for TBox Reasoning. In: Proc. of AAAI 2010 (2010)
13. Sattler, U., Schneider, T., Zakharyaschev, M.: Which kind of module should I extract? In: Proc. of DL 2009 (2009)