

# The Justificatory Structure of the NCBO BioPortal Ontologies

Samantha Bail, Matthew Horridge, Bijan Parsia, Ulrike Sattler

The University of Manchester  
Oxford Road, Manchester, M13 9PL  
{bails,bparsia,sattler@cs.man.ac.uk}

**Abstract.** Current ontology development tools offer debugging support by presenting justifications for entailments of OWL ontologies. While these minimal subsets have been shown to support debugging and understanding tasks, the occurrence of *multiple* justifications presents a significant cognitive challenge to users. In many cases even a single entailment may have many distinct justifications, and justifications for distinct entailments may be critically related. However, it is currently unknown how prevalent significant numbers of multiple justifications per entailment are in the field. To address this lack, we examine the justifications from an independently motivated corpus of actively used biomedical ontologies from the NCBO BioPortal. We find that the majority of ontologies contain multiple justifications, while also exhibiting structural features (such as patterns) which can be exploited in order to reduce user effort in the ontology engineering process.

## 1 Introduction

Debugging and repair of an OWL ontology is a crucial step in the ontology development process in order to ensure the correctness and quality of the ontology. Finding the source of an error and modifying it to remove the fault can be a tedious and error-prone task in large and often complex OWL ontologies. Adequate *explanation* support for arbitrary entailments is therefore an essential component of OWL ontology editors.

Justifications, minimal subsets of an ontology that are sufficient for an entailment to hold, are currently the prevalent form of explanation in OWL ontology development tools such as Protégé 4. Previous research has mainly dealt with improving the comprehensibility of single justifications for an individual entailment [16,8,13], as well as optimising the performance of computing justifications [14,3,24]. We are now attempting to tackle the issue of coping with *multiple justifications*.

Multiple justifications for a single entailment occur in a large number of OWL ontologies, regardless of the size or description logic expressivity of the ontology, often reaching up to several hundred justifications per entailment [4]. However, even small numbers of multiple justifications can cause a cognitive overload for the ontology engineer. Choosing a *minimal repair*, i.e. a smallest

possible modification to remove the entailment without affecting the remainder of the ontology, requires significant cognitive effort from the user when faced with not just one, but multiple justifications.

Further, repairing entailments in isolation might also cause non-minimal repairs, as demonstrated in [15]. Considering multiple entailments to repair simultaneously almost certainly<sup>1</sup> requires the user to deal with multiple justifications. As in the case of multiple justifications for a single entailment, the ontology engineer must recognize *relationships* between the justifications in order to find a suitable repair. While the root and derived justifications described in [15] point out one type of relations (namely subset relations), there exist many other structural aspects of justifications which have not been explored yet.

While there is a clear use case for improved coping mechanisms, we may also consider gathering additional knowledge about an ontology from the justifications occurring in it. A user may want to learn about the modelling of an ontology by considering not only its explicitly asserted structure and metrics, but also its implicit structure, which is described by the relations between entailments and their entailing axiom sets.

To date, there has been no systematic investigation into the problem of multiple justifications in an independently motivated corpus of OWL ontologies. In this paper, we analyse the relationships between justifications in a set of ontologies from the biomedical domain which were extracted from the NCBO BioPortal.<sup>2</sup> This analytical work constitutes the first step on the road to developing an explanation tool with improved coping strategies for multiple justifications with the aim of supporting ontology engineers in the debugging process.

To facilitate the description of justificatory structure, we introduce a graph-based framework for capturing and analysing relationships between justifications in OWL ontologies. Using these JGraphs, we outline different aspects of the justificatory structure and perform an analysis of an ontology corpus. The contributions of this paper are: 1) A framework to compute and describe the justificatory structure as the foundation for improved explanation support for multiple justifications. 2) Metrics for OWL ontologies that describe implicit features of the ontology. 3) A survey of a representative set of bio-ontologies that demonstrates the concept of justificatory structure and provides insight into structural aspects of the corpus.

## 2 Preliminaries

In the following section we provide a brief overview of the Web Ontology Language OWL and discuss the notion of entailment sets. We then introduce justifications as a form of explanation for entailments.

---

<sup>1</sup> A set of axioms can be a minimal entailing set for multiple entailments.

<sup>2</sup> <http://biportal.bioontology.org>

## 2.1 OWL

The Web Ontology Language OWL 2,<sup>3</sup> may be regarded as a syntactic variant of the expressive description logic (DL) [2]  $\mathcal{SROIQ}$ , with an OWL 2 ontology corresponding to a set of  $\mathcal{SROIQ}$  [11] axioms. These axioms, i.e. statements about the entities in the ontology, can take the form of subsumptions (denoted by the symbol  $\sqsubseteq$  in DL and `SubClassOf` in OWL Manchester Syntax<sup>4</sup>) and equivalent classes (denoted by  $\equiv$  in DL and `EquivalentClasses` in Manchester Syntax). They may involve complex class expressions to describe the relationships between the classes in the ontology, which are based on a wide range of available constructors in OWL 2. For example, the equivalent class axiom

$$\text{DNA} \equiv \text{NucleicAcid} \sqcap \exists \text{hasPart.Deoxynucleotide}$$

defines the class DNA as a NucleicAcid that has some (at least one) part which is a Deoxynucleotide. In addition to subsumptions and equivalences between classes, we can also make statements about the *individuals* and the *roles* in the ontology. In the remainder of this paper we will use the term OWL interchangeably with OWL 2 when referring to OWL 2 ontologies.

## 2.2 Entailments

Any statement which holds in all models of an ontology  $\mathcal{O}$  is considered an *entailment* of the ontology. For example, an ontology  $\mathcal{O}$  containing the above axiom entails that DNA is a subclass of NucleicAcid, which is expressed as  $\mathcal{O} \models \text{DNA} \sqsubseteq \text{NucleicAcid}$ .

We consider the entailment set of an ontology to be a set of entailments of interest, given by a function  $\varepsilon(\mathcal{O})$ :

**Definition 1 (Entailment set)** *Let  $\mathcal{O}$  be an ontology and  $\varepsilon(\mathcal{O})$  a function that returns a finite set of axioms  $\{\eta_1 \dots \eta_n\}$  such that  $\mathcal{O} \models \eta_i$ ; this set is an entailment set of  $\mathcal{O}$ .*

While the entailment relation  $\mathcal{O} \models \eta$  is well defined through the semantics of description logics, the term is often used in an ambiguous way [6]. In order to specify a particular *finite* subset of the set of all entailments of an ontology  $\mathcal{O}$ , multiple variables need to be fixed: For the purpose of analysing the justificatory structure of ontologies, we focus on entailments that are *direct subsumptions* between atomic (named) classes, including  $\top$  and  $\perp$ . This set includes asserted entailments, as there may be additional (other than the axiom itself) reasons for the entailment to hold, which may be missed when excluding asserted entailments from the analysis. Tautologies such as  $A \sqsubseteq A$ ,  $A \sqsubseteq \top$  and  $\perp \sqsubseteq A$  for a named class  $A$  are omitted, as they hold no information value.

<sup>3</sup> <http://www.w3.org/TR/2009/REC-owl2-overview-20091027>

<sup>4</sup> <http://www.w3.org/TR/owl2-manchester-syntax>

### 2.3 Justifications

Justifications [23,20] are a form of explanation of entailments of OWL ontologies, which is used in OWL ontology editors such as Protégé 4 to provide explanation support to the user. A justification is a minimal subset of an ontology  $\mathcal{O}$  that causes an entailment  $\eta$  to hold.

**Definition 2 (Justification)**  $\mathcal{J}$  is a justification for  $\mathcal{O} \models \eta$  if  $\mathcal{J} \subseteq \mathcal{O}$ ,  $\mathcal{J} \models \eta$  and, for all  $\mathcal{J}' \subset \mathcal{J}$ , it holds that  $\mathcal{J}' \not\models \eta$ .

For every axiom which is asserted in the ontology, the axiom itself naturally is a justification. We are, however, only interested in *non-trivial entailments*, i.e. justifications which have some (at least one) justification which is not the axiom itself; in this case, the justification is also called a *non-trivial justification*.

A justification is defined with respect to a single entailment  $\eta$  and an ontology  $\mathcal{O}$ ; in order to describe the set of all justifications for all entailments in an entailment set  $\varepsilon(\mathcal{O})$ , we introduce the notion of *justification sets*:

**Definition 3 (Justification set)** Given an ontology  $\mathcal{O}$  and a function  $\varepsilon(\mathcal{O})$ , the justification set  $Justs(\mathcal{O}, \varepsilon)$  is the set of all justifications  $\{\mathcal{J}_1 \dots \mathcal{J}_m\}$ ,  $\mathcal{J}_i \subseteq \mathcal{O}$ , for the axioms in the entailment set  $\varepsilon(\mathcal{O})$ .

Further, we define the set of all axioms occurring in all justifications for a particular entailment set:

**Definition 4 (Justification axioms)**

$$JustAx(\mathcal{O}, \varepsilon) = \{\alpha \mid \text{there is a } \mathcal{J} \in Justs(\mathcal{O}, \varepsilon) \text{ s.t. } \alpha \in \mathcal{J}\}$$

With respect to debugging unwanted entailments (i.e.  $\varepsilon(\mathcal{O})$  is the set of all unwanted entailments, e.g. unsatisfiable classes) of an ontology, a *repair* is a subset of  $JustAx(\mathcal{O}, \varepsilon)$  which, if removed from the ontology, would *break* all these unwanted entailments. We are particularly interested in finding a *minimal repair*, which corresponds to a *minimal hitting set* [21] for the justifications in  $Justs(\mathcal{O}, \varepsilon)$ .

## 3 JGraphs

In this section we provide the necessary definitions for justification graphs which capture the relations between axioms in the ontology, justifications (sets of axioms) and entailments of interest. Based on the above definitions of entailment sets and justification sets, we can now define the justification graph of an ontology  $\mathcal{O}$ . A justification graph (JGraph)  $G_{\mathcal{J}}$  is a directed graph whose set of vertices is the union of the set of axioms  $\varepsilon(\mathcal{O})$  which are entailed by  $\mathcal{O}$  and the set  $JustAx(\mathcal{O}, \varepsilon)$  of axioms that participate in justifications for these entailments, together with the set of all justifications  $Justs(\mathcal{O}, \varepsilon)$ . The edges indicate whether an axiom is an element of a justification, and whether a justification is a justification for a particular entailed axiom; hence, the graph is bipartite.

**Definition 5 (Justification graph)**

$$G_{\mathcal{J}} = (\varepsilon(\mathcal{O}) \cup JustAx(\mathcal{O}, \varepsilon) \cup Justs(\mathcal{O}, \varepsilon), E_1 \cup E_2) \text{ where}$$

$$E_1 = \{(u, v) \in \varepsilon(\mathcal{O}) \cup JustAx(\mathcal{O}, \varepsilon) \times Justs(\mathcal{O}, \varepsilon) \mid u \in v\},$$

$$E_2 = \{(v, w) \in Justs(\mathcal{O}, \varepsilon) \times \varepsilon(\mathcal{O}) \mid v \in Justs(\mathcal{O}, w)\}.$$

Side remarks: 1) The set of justifications for a given set of entailments is unique, so is the set of axioms in the justifications, and the edges follow from these unambiguous relations; therefore the JGraph is unique. 2)  $\varepsilon(\mathcal{O})$  and  $JustAx(\mathcal{O}, \varepsilon)$  are not disjoint; i.e. an axiom in  $\mathcal{O}$  may have a non-trivial justification, while also being an element of a justification for some other entailment. 3) Any axiom vertex in the graph with an in-degree of at least one is an entailment in  $\varepsilon(\mathcal{O})$ . (4) Similarly, any axiom vertex in the graph with an out-degree of at least one is in  $JustAx(\mathcal{O}, \varepsilon)$ .

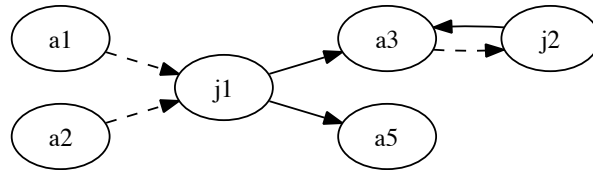
The principle of JGraphs is demonstrated by the following example ontology:

$$\begin{aligned} \mathcal{O} = \{ & A \sqsubseteq \exists R.B, & (a1) \\ & \exists R.B \sqsubseteq C \sqcap D, & (a2) \\ & A \sqsubseteq D, & (a3) \\ & F \sqsubseteq G \} & (a4) \end{aligned}$$

The entailment set  $\varepsilon(\mathcal{O})$  comprising the direct and indirect atomic subsumptions that are entailed by  $\mathcal{O}$  contains the following axioms:

$$\begin{aligned} \varepsilon(\mathcal{O}) = \{ & A \sqsubseteq C, & (a5) \\ & A \sqsubseteq D, & (a3) \\ & F \sqsubseteq G \} & (a4) \end{aligned}$$

Only the first two entailments  $a5$  and  $a3$  in  $\varepsilon(\mathcal{O})$  have a non-trivial justification  $j1 = \{a1, a2\}$ , and the entailed axiom  $a3$  has an additional trivial justification, which is the axiom itself:  $j2 = \{a3\}$ . The set of vertices in JGraph  $G_{\mathcal{J}}$  therefore is:  $\{a1, a2, a3, a5, j1, j2\}$ . The respective sets of edges in the graph are:  $E_1 = \{(a1, j1), (a2, j1), (a3, j2)\}$  and  $E_2 = \{(j1, a5), (j1, a3), (j2, a3)\}$ .



**Fig. 1.** Example of a JGraph

## 4 Justificatory Structure

Using the JGraphs defined above, we outline different aspects of the justificatory structure, which allows us to examine and describe implicit structural properties of an OWL ontology.

### 4.1 Number and Size of Justifications

**Justificatory Redundancy** The number of justifications per entailment is an indicator of *justificatory redundancy* in the ontology; it demonstrates ‘how often the same thing is expressed in different ways’. The in-degree of entailments in the JGraph corresponds to this metric. We must point out that, while the term redundancy has mostly negative connotations, this measurement can also be a criterion for the *richness* and *inferential power* of an ontology. It is also not to be confused with logical redundancy, as this would imply that it is possible to remove a set of axioms from the ontology without breaking any entailments. Furthermore, we cannot make any claims about the purpose of the axioms in the justifications; they might have been added without the intention of causing the entailment.

**Activity** Finally, with respect to the user effort required when dealing with an ontology, we define the *activity* of an OWL ontology. This is the total number of axioms that occur in justifications for non-trivial entailments, that is, the size of the subset of the ontology which actively participates in inference. An ontology in which the set of asserted axioms is the same as the set of inferred ones (with respect to some definition of  $\varepsilon(\mathcal{O})$ ) has an activity value of 0. This measurement does not take into account equivalent sets of axioms where the axioms have different sizes; for example, a subsumption of the form  $A \sqsubseteq B \sqcap D$ ,  $A \sqsubseteq C \sqcap D$  could be re-written into a single axiom  $A \sqsubseteq B \sqcap C \sqcap D$ , which is logically equivalent, but has a lower activity value.

### 4.2 Self-Justifications

Any justification which is simply the entailed axiom itself is classified as a self-justification. In the JGraph this is expressed as a cycle between an axiom node and a justification node, where the in-degree of the justification node is one. The justification  $j_2$  in Figure 1 illustrates a self-justification for the axiom  $a_3$ . There are different reasons for the existence of self-justifications: (1) A conscious design decision to improve reasoner performance or a tailoring towards a particular ontology browser interface which does not support reasoning, i.e. the inferred subsumptions were added back into the ontology. (2) The absence of a reasoner during the ontology engineering process and the modeller not being aware that the subsumption is already entailed. (3) The entailed subsumption could simply be a side-effect of axioms that were added to the ontology without the aim of causing the entailment.

### 4.3 Justification Overlap

**Arbitrary Overlap** Justifications for both single and multiple entailments may share some axioms, i.e. the justifications *overlap* to a certain extent. With respect to coping with multiple justifications, this overlap is an indicator for a common *lemma*, i.e. an intermediate entailment caused by a subset of a justification [9]. A suitable lemma may support understanding structural similarities between multiple justifications, which in turn reduces the task of understanding multiple seemingly distinct justifications to understanding a smaller number of lemmas.

**Root and Derived Justifications** A special case of justification overlap are *root* and *derived* unsatisfiable classes [15], which describes the containment of one justification in another. An unsatisfiable class  $C$  is *derived* if one of its justifications is a superset of the justification of another unsatisfiable class  $D$ ; all other unsatisfiable classes are *roots*. Meyer et al. [17] propose a repair strategy that extends root and derived unsatisfiable classes to arbitrary sets of entailments. These relationships are captured by the edge set  $E_3$  in the JGraph:

**Definition 6 (Root and derived justifications)** *Given a JGraph  $G_{\mathcal{J}} = (\varepsilon(\mathcal{O}) \cup JustAx(\mathcal{O}, \varepsilon) \cup Justs(\mathcal{O}, \varepsilon), E_1 \cup E_2)$  and a vertex  $v \in Justs(\mathcal{O}, \varepsilon)$ ;  $v$  is a derived justification if there is a justification  $v'$  such that  $\{w \mid (w, v) \in E_1\} \supset \{w \mid (w, v') \in E_1\}$ . Else,  $v$  is a root justification.*

This definition aligns with the root and derived unsatisfiable classes defined in [15]; in this case, the entailment set  $\varepsilon(\mathcal{O})$  is comprised of all unsatisfiable classes in an ontology  $\mathcal{O}$ .

**Equality** Equality is another special case of justification overlap, where the justifications for different entailments contain the same axioms. In terms of the JGraph, these justifications are represented by a single vertex which has an out-degree greater than one. The equality of justification is illustrated by the following two axioms from the above example ontology:

**Example 1 (Multiple Entailments)**

$$\begin{aligned} A &\sqsubseteq \exists R.B \\ \exists R.B &\sqsubseteq C \sqcap D \end{aligned}$$

This minimal set of axioms entails the two atomic subsumptions  $A \sqsubseteq C$  and  $A \sqsubseteq D$  and therefore represents a justification for both entailments. In the graph shown in Figure 1, this justification is represented by the vertex labelled  $j1$ , which entails the two axioms  $a3$  ( $A \sqsubseteq D$ ) and  $a5$  ( $A \sqsubseteq C$ ).

We may consider the number of axioms that a justification entails as the *inferential power* of the justification, answering the question ‘how much can be expressed with how little?’. In the context of repairing unwanted entailments, users can benefit from examining multiple unwanted entailments at the same time if they share some justifications rather than looking at each justification in isolation, as this reduces the total number of justifications to repair.

**Axiom Power** The *power* of an axiom, also denoted as *arity* [23], is the out-degree of any axiom in  $JustAx(\mathcal{O}, \varepsilon)$ . The respective axiom occurs in multiple justification, which corresponds to a justification overlap of size one. *Key axioms* are those axioms with the maximal out-degree. These provide informations about the ontology in two ways: With respect to repairing an unwanted entailment, the user may focus on removing or weakening the key axioms first, as they have the highest repair powers. Secondly, key axioms are those statements that contribute to a large number of entailments of the ontology, and are therefore structurally (not necessarily in the context of domain knowledge) relevant to understanding the ontology.

A set of axioms can be a justification for more than one entailment. Therefore, we need to consider how many entailments would be removed from the ontology through the removal or weakening an axiom in the repair process. The *impact* of an axiom is the number of entailments that would break in addition to the entailment we focus on, which corresponds to the out-degree of all justifications that the axiom has edges to.

#### 4.4 Patterns

Two types of patterns can be identified in the context of the justificatory structure: (1) Graph surface patterns, and (2) isomorphism between justifications. A surface pattern is a structural similarity in the JGraph, such as matching subgraphs. Surface patterns in the JGraph reveal modelling similarities in the ontology, regardless of whether the justifications and axioms in the pattern also interact in a similar way. Highlighting a pattern of this type may support user understanding of the modelling in the ontology, while it may also be an indicator for isomorphic justifications.

Two justifications  $\mathcal{J}_1$  and  $\mathcal{J}_2$  are isomorphic [7] if there is an injective renaming from  $\mathcal{J}_1$  to  $\mathcal{J}_2$ , i.e. the axioms in the justifications have the same structure while using different class and property names. It can be claimed that if a person is able to understand  $\mathcal{J}_1$  (from a structural point of view, not considering domain knowledge), they can also understand  $\mathcal{J}_2$ . Making this sameness explicit may reduce the user effort required when faced with a large number of justifications in the debugging process, as it reduces the number of justifications that need to be examined.

#### 4.5 Components

The number of components of the graph provides a measure for the disjointness of justifications in the whole ontology. The disjointness of justifications strongly affects the justification computation process, which makes use of Reiter’s Hitting Set Tree (HST) algorithm for diagnosis [21,22]. In the HST, the vertices are labelled with justifications and the paths constitute hitting sets, i.e. minimal repairs for the justifications. Optimisations for the HST algorithm are mainly based on closing a branch in the tree if the path to it is labelled with a superset of an existing path, which is not possible if the justifications are disjoint. This



leads to a rapid growth of the HST and has significant negative effects on the performance of computing all justifications for an entailment.

## 5 BioPortal Ontology Survey

In this section we apply the JGraph metrics on a corpus of ontologies used in bio-health applications. By analysing selected aspects of justificatory structure, we draw conclusions regarding the occurrence and the nature of multiple justifications in OWL ontologies. In a nutshell, the ontologies surveyed cover a wide range of justificatory structure from very lean to very rich, with size or DL flavour of the ontology being no indicator of its implicit structure.

### 5.1 Test Corpus: The NCBO BioPortal

The purpose of this study was to analyse the justificatory structure of a realistic and representative set of OWL ontologies. Thus, the test corpus was selected based on the following criteria: 1) The ontologies had to be publicly accessible. 2) In order to avoid hand-picking ‘suitable’ ontologies that may not be representative of naturally occurring ontologies, the choices were between a random sample of web ontologies, or an existing set from a web-based repository. 3) For the same reason, we focused on realistic ontologies that were actively used, i.e. ontologies that were built simply for training purposes (such as the *Koala* or *Pizza* ontologies) would be excluded. The NCBO BioPortal [18] repository meets all the above criteria while containing a large number of OWL and OWL-compatible ontologies; it was therefore selected as the test corpus for our survey.

The BioPortal provides ontologies from various groups from the biomedical domain, including the full set of daily updated OBO Foundry<sup>5</sup> ontologies, which are built based on common design principles. OBO ontologies use a flat-file format, which can be translated into OWL 2 and were therefore included in the test corpus.

**Dataset** At the time of downloading (12 March 2011), the BioPortal repository listed 226 latest versions of ontologies in OWL and OBO format, out of which 218 could be downloaded and parsed with the OWL API<sup>6</sup> parsers [10]. 8 ontologies could not be processed due to the file being not available under the given URL, or parsing errors. For each ontology the imports closure was downloaded and merged with the root ontology, while missing imports were ignored. We then extracted all entailed atomic subsumptions from the parseable OWL files and excluded those ontologies that did not contain any non-trivial entailments, which left us with 72 ontologies. At this stage, 5 ontologies were removed from the set as they could not be processed by the justification generation and the JGraph framework due to their large size and number of entailments.

<sup>5</sup> <http://obofoundry.org>

<sup>6</sup> <http://owlapi.sourceforge.net>

The structural analysis of the remaining 67 ontologies was further restricted to only include ontologies with coherent TBoxes, i.e. ontologies that contained only statements about the concept hierarchy and no unsatisfiable classes. Both justifications for unsatisfiable classes and ABox entailments need to be treated differently from justifications for subsumptions between named classes; a separate investigation of the justificatory structure of both incoherent ontologies and ontologies with ABoxes are omitted due to space limitations, but are part of future work. This filtered out 20 ontologies with an ABox, and 5 ontologies with unsatisfiable classes, leaving us with 42 ontologies that contained non-trivial entailments for which the justifications could be generated and processed in the JGraph generator.

**Imported Entailments** In the next stage of data pruning, the import structure of the 42 ontologies was examined in order to determine to which extent entailments were imported from external ontologies. An entailment whose justifications contain only axioms from an imported ontology is classified as an ‘imported entailment’, whereas an entailment whose justifications contain only axioms from the importing ontology is called a ‘native entailment’ [6]. 28 ontologies did not have any imported entailments at all, which could be either due to them having no imports, the imported ontology having no entailments that matched our criteria, or missing imports, which were ignored in the pre-processing stage.

We found that 7 ontologies in BioPortal import the Basic Formal ontology (BFO), an ‘upper’ ontology for biological data which itself is contained in the BioPortal corpus. 3 of these ontologies had 70 imported entailments each, which all stemmed exclusively from BFO, and no native entailments. The remaining 4 ontologies had the 70 imported entailments from BFO, plus additional entailments which were either native or imported from ontologies other than BFO. A further 7 ontologies had imported entailments from other ontologies, which, in most cases, could be attributed to the ontology intentionally being split up over several files. For example, the Chemical Information ontology (76 entailments) had 1 native entailment, 72 imported entailments from an ontology titled ‘cheminf-external’, and 3 entailments from ‘cheminf-core’.

In order to prevent skewed results due to the dominance of BFO, the ontologies which imported BFO were also removed from the set (while BFO itself remained in the corpus). This left us with 35 ontologies which had only native (or mixed) non-trivial entailments from a coherent TBox, which could be processed by the JGraph framework.<sup>7</sup>

---

<sup>7</sup> Due to space limitations and the wealth of data obtained in the experiments, we have to omit tables and graphs displaying the results. The raw data from the graph analysis as well as detailed overview tables are available from <http://owl.cs.manchester.ac.uk/research/publications/supporting-material/iswc2011-juststruct>.

## 5.2 Results and Analysis

**Entailments** From the 35 ontologies, 12,010 non-trivial entailments were extracted, with a total of 7,176 distinct non-trivial justifications. In addition, 2,340 self-justifications were found, i.e. 2,340 entailments were asserted in the ontologies while also having additional justifications. The average number of non-trivial entailments across all ontologies is 343, which is mainly affected by the large number of entailments in the NCI Thesaurus (7,862), and the Experimental Factor ontology (1,787), which deviates by an order of magnitude from the other ontologies.

The majority of ontologies (74.3%) that produce non-trivial entailments has 100 or less entailments. Surprisingly, small numbers of non-trivial entailments occurred in some of the largest ontologies with high DL expressivity, such as the MaHCO ontology ( $\mathcal{ALCHIQ}(\mathcal{D})$ , 13,844 axioms, 26 non-trivial entailments) and Cardiac Electrophysiology ( $\mathcal{SHF}(\mathcal{D})$ , 176,113 axioms, 19 non-trivial entailments).

As we can see from these examples, the size of an ontology does not necessarily affect the number of non-trivial entailments. This is confirmed by the Spearman’s rank coefficient<sup>8</sup> of  $\rho = 0.18$ , which indicates that there is no correlation between the two values.

**Multiple Justifications** The average in-degree for the JGraph vertices representing entailments is 1.3 (standard deviation  $\sigma = 4.4$ ), not counting the self-justifications (i.e. the justification which is the asserted axiom itself). In particular, only 10 out of the 35 ontologies (28.6%) have exactly one justification for each of its entailments;<sup>9</sup> 27 ontologies (57.1%) have between one and 2 justifications per entailment; and 5 (14.3%) ontologies have an average of more than 2 justifications per entailment.

The largest number of justifications for a single entailment (236) can be found in the Gene Ontology Extension, an ontology that has several entailments with more than 100 justifications each, followed by the Experimental Factor ontology, which has 20 entailments with 16 or more justifications each. There exists no correlation between the size of an ontology and the number of justifications per entailments ( $\rho = -0.21$ ), so neither size nor expressivity of an ontology are an indicator for multiple justifications.

The occurrence of multiple justifications in 71.4% of the surveyed ontologies, as well as the large maximal number of justifications per entailment in some ontologies shows that both the computational requirements as well as the cognitive complexity of multiple justifications can pose a significant challenge when dealing with OWL ontologies found in practice.

---

<sup>8</sup> Values for Spearman’s rank coefficient range from -1, which indicates a perfect negative correlation between two variables, to +1 (perfect positive correlation), with 0 indicating no correlation.

<sup>9</sup> Only 11 out of the 800 justifications in the NCI Thesaurus have multiple justifications (2 and 3 respectively), which yields an average of 1 justification per entailment after rounding.

**Self-Justifications** In the surveyed corpus, 2,340 of the 12,010 entailments have self-justifications, which means that the entailments are asserted as well as inferred through additional reasons in the ontology. While the occurrence of self-justifications is common (e.g. due to the entailment simply being a side-effect of some axioms in the ontology), it is surprising that 8 ontologies do not contain any self-justifications, despite large numbers of both entailments and justifications. On the other extreme, one ontology, the Software Ontology, has a self-justification for each of its 332 entailments; a detailed discussion of this ontology follows in Section 5.2.

**Size of Justifications** The average size of justifications in the ontology is 5.9 axioms ( $\sigma = 4.0$ ), with a maximum size of 20 axioms for a justification in the NCI Thesaurus. We have to emphasize that the justifications were only computed for direct subsumptions; this means that large justifications cannot stem from long subsumption chains. The largest justifications were found in the Gene Ontology Extension (average size: 9.5 axioms) and the NCI Thesaurus (9.6 axioms), with the latter having over 300 entailments that have justifications of size 10 or larger.

All 35 ontologies have an average justification size of less than 10 axioms, which gives us a useful indicator of the potential cognitive complexity of understanding justifications: While there are some extreme cases, such as the NCI Thesaurus, most ontologies produce justifications with a size that can be deemed ‘manageable’ by human users. As with the number of justifications per entailment, we cannot detect any correlation between the size of an ontology and the average size of its justifications ( $\rho = 0.11$ ).

**Activity** Regarding the number of axioms of the ontology that participate in justifications, the largest total numbers can be found in the NCI Thesaurus (6,479 axioms) and the Experimental Factor ontology (3,813 axioms). Interestingly, when taking into account the total size of the ontology, the NCI Thesaurus only uses 4.4% of its axioms in justifications, whereas the axioms occurring in justifications in Experimental Factor make up 53.7% of the whole ontology, with both ontologies having relatively large numbers of entailments (7,862 and 1,787 entailments respectively). This state is reflected by the average number of entailments per justification, i.e. the inferential power of the justifications in the ontology: While the NCI Thesaurus has an average of 9.8 entailments per justification, the justifications in Experimental Factor only have 1.2 entailments each.

The majority of ontologies in the corpus (21 out of 35) have comparatively small numbers (less than 100) of axioms that participate in inference, regardless of the size of the ontology. Not surprisingly, there is a strong correlation between the total number of non-trivial entailments of an ontology and the number of axioms that participate in the justifications for these entailments ( $\rho = 0.9$ ).

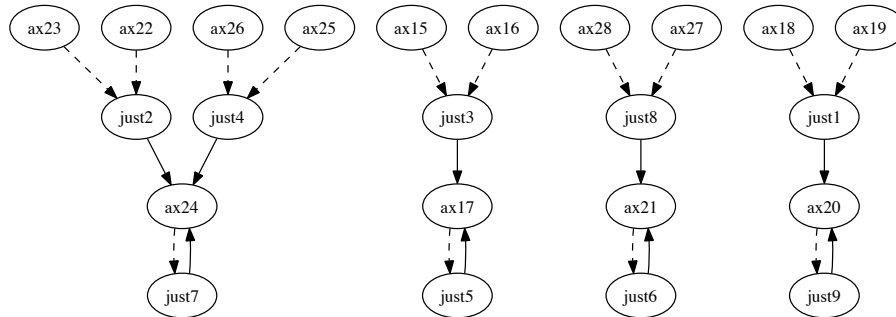
**Axiom Power** The average axiom power across all ontologies in the corpus is 3.0, which means that, on average, an axiom occurs in 3 justifications. While only

4 ontologies have an average axiom power of exactly 1.0, all remaining ontologies have surprisingly high-power axioms. More than half of the ontologies (54.3%) contain at least one axiom which occurs in 9 or more justifications, and 31.4% have axioms with a power of 20 or higher, peaking at one axiom that occurs in 510 justifications in the Experimental Factor ontology.

Again, this shows that considering justifications in isolation when debugging an ontology can lead to non-minimal repairs, e.g. by removing a different axiom from each justification rather than removing a shared axiom, which also causes an unnecessary overhead in terms of effort required.

Since the analysis of impact depends on a particular justification/entailment pair in the context of ontology repair, a general discussion on this metric is omitted.

**Patterns** One interesting case of patterns in the form of isomorphic justifications is the Software ontology (2,080 axioms, DL expressivity  $\mathcal{ALCHLN}$ ). Almost all of the 332 entailments have exactly one self-justification and one additional non-trivial justification which contains 2 axioms; in fact, only one of the justifications contains 3 axioms. 318 of these entailments are of the form  $X \sqsubseteq R \text{ Software}$  for  $X$  being some class name in the ontology, with all of the justifications of size two being structurally isomorphic and the equivalence axiom occurring in all 318 justifications:  $\mathcal{J}_i = \{R \text{ Software} \equiv \exists is\_encoded\_in.R \text{ Language}, X \sqsubseteq \exists is\_encoded\_in.R \text{ Language}\}$ .<sup>10</sup>



**Fig. 2.** JGraph of the Biopax ontology, with 3 isomorph subgraphs

A similar pattern can be found in the Biopax ontology (Figure 2). The 3 isomorphic subgraphs are justifications for atomic subsumptions of the type  $A \sqsubseteq B$  which themselves are asserted in the ontology (i.e. they have self-justifications,  $just5$ ,  $just6$  and  $just9$ ). 2 of the 3 justifications ( $just3$  and  $just8$ ) contain an axiom of the type  $A \sqsubseteq \exists d.x$ , where  $d$  is a data property and  $x$  a data property value (in this case a string), together with a data property domain

<sup>10</sup> The ontology uses IDs for the class names, which we here display with their human-readable labels instead.

axiom  $Domain(d, B)$ . Justification *just1* contains axioms of the same structure with an object property and object property domain axiom instead. These 3 justifications have exactly the same surface structure in the graph, while not being strictly isomorphic.

This phenomenon highlights the importance of structural analysis of multiple justifications, as we can easily understand the nature of several hundred justifications (as in the case of the Software ontology) by examining only one justification and understanding the isomorphic structure, rather than dealing with every single justification independently.

## 6 Related Work

Structural analysis of (OWL) ontologies is an actively researched topic in the areas of ontology quality measurement, ontology integration, and ontology matching. Most of the frameworks developed for these purposes focus on analysing the asserted class hierarchy using basic metrics, such as the numbers of root classes, numbers of leaf classes and depth of the inheritance tree, to determine structural metrics of the ontology [25,27,1,19]. A more in-depth approach to analysing the asserted class graph is proposed in [12], which applies methods from Social Network Analysis for the purpose of comparing ontologies.

The research area of exploring structural relationships between the *reasons* for entailments of OWL ontologies is in its early stages. In one of the first approaches to analysing this *implicit* structure of ontologies, Vrandečić et al. [26] introduce ontology metrics that also consider the semantics of the ontology language, mainly focusing on the entailed statements of an ontology.

*Root* and *derived* justifications [15] provide a way of exploiting structural relations between justifications for multiple entailments. Focusing on root justifications can drastically reduce the number of justifications a user has to examine when attempting to repair unsatisfiable classes. In [17], the authors describe an extension for root and derived justifications to cover *arbitrary* sets of entailments beyond unsatisfiable classes. The focus of the work of root and derived justifications is to provide improved ontology debugging and repair support; there is no indication of inferring information about the ontology from its root and derived structure.

In [5], we introduce the notion of justificatory structure and conduct a preliminary survey on a set of ontologies. We analyse a small number of properties, such as the number of justifications per entailment, with findings indicating that the problem of multiple justifications for entailments is common in real-life OWL ontologies.

The characteristics of justifications in bio-ontologies in particular are the focus of a survey of the NCBO BioPortal [10]. In this study, the number of non-trivial entailments and justifications per entailment are analysed against the background of the ontology size and expressivity. It is found that a large number of ontologies contains non-trivial entailments, which indicates the use of inference in the ontology engineering process.

## 7 Conclusion and Future Work

We have presented a graph-based framework that captures the relations between axioms, entailments and justifications of an OWL ontology. Analysing this *justificatory structure* of OWL and OWL-compatible ontologies from the NCBO BioPortal has clearly shown that multiple justifications do occur in a large proportion (71.4%) of the surveyed ontologies.

Furthermore, some ontologies have very large numbers of multiple justifications per entailment (up to several hundred), which poses a significant computational and cognitive challenge against the background of explanation support for ontology debugging tasks. We have found no correlations between the DL expressivity and size of an ontology and the complexity of its justificatory structure. With improved ontology development tool support and modelling patterns, we expect OWL ontologies to become more complex in the future, which may lead to a more complex justificatory structure. We can therefore conclude that it is necessary to focus attention on developing improved coping mechanisms for multiple justifications in order to reduce user effort and limit the computational load.

In our presentation of justificatory structure, we have also discussed and demonstrated structural relations between the justifications in an ontology, such as axiom power, patterns and justification overlap, which may be exploited for these coping mechanisms. Based on the axiom power analysis and the patterns found in the surveyed ontologies (e.g. the Software ontology), we have shown how structural analysis can drastically reduce the number of seemingly distinct justifications that have to be examined when attempting to understand the justifications in an ontology.

For future work, we plan to further explore justification overlap and lemmas, using approaches from formal concept analysis. This will provide us with the necessary structural information which can then be used to suggest to the user a suitable repair strategy when confronted with multiple justifications. Finally, as the current basic visualization of JGraphs is clearly limited in terms of scalability and the representation of most structural aspects, we aim to investigate approaches to developing a user oriented interaction framework.

## References

1. H. Alani and C. Brewster. Metrics for ranking ontologies. In *Proc. of EON-06*, 2006.
2. F. Baader, D. Calvanese, D. McGuinness, P. Patel-Schneider, and D. Nardi. *The description logic handbook: theory, implementation, and applications*. Cambridge University Press, 2003.
3. F. Baader, R. Penaloza, and B. Suntisrivaraporn. Pinpointing in the description logic EL+. *Proc. of AAAI-07*, 4667:52–67, 2007.
4. S. Bail, B. Parsia, and U. Sattler. Justbench: A framework for OWL benchmarking. In *Proc. of ISWC-10*, 2010.
5. S. Bail, B. Parsia, and U. Sattler. The justificatory structure of OWL ontologies. In *Proc. of OWLED-10*, 2010.

6. S. Bail, B. Parsia, and U. Sattler. Extracting finite sets of entailments from OWL ontologies. In *Proc. of DL-11*, 2011.
7. M. Horridge, S. Bail, B. Parsia, and U. Sattler. The cognitive complexity of OWL justifications. In *Proc. of DL 2011*, 2011.
8. M. Horridge, B. Parsia, and U. Sattler. Laconic and precise justifications in OWL. In *Proc. of ISWC-08*, pages 323–338, 2008.
9. M. Horridge, B. Parsia, and U. Sattler. Lemmas for justifications in OWL. In *Proc. of DL 2009*, 2009.
10. M. Horridge, B. Parsia, and U. Sattler. The state of bio-ontologies. In *Proc. of ISMB-11*, 2011.
11. I. Horrocks, O. Kutz, and U. Sattler. The even more irresistible SROIQ. In *Proc. of KR-06*, pages 57–67, 2006.
12. B. Hoser, A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. Semantic network analysis of ontologies. In *Proc. of ESWC-06*, pages 514–529. Springer, 2006.
13. Q. Ji, G. Qi, and P. Haase. A relevance-based algorithm for finding justifications of DL entailments. Technical report, University of Karlsruhe, 2008.
14. A. Kalyanpur, B. Parsia, M. Horridge, and E. Sirin. Finding all justifications of OWL DL entailments. *Proc. of ISWC/ASWC-07*, 4825:267, 2007.
15. A. Kalyanpur, B. Parsia, E. Sirin, and J. Hendler. Debugging unsatisfiable classes in OWL ontologies. *J. of Web Semantics*, 3(4):268–293, 2005.
16. J. Lam, J. Pan, D. Sleeman, and W. Vasconcelos. A fine-grained approach to resolving unsatisfiable ontologies. In *Proc. of WI-06*. Springer, 2006.
17. T. Meyer, K. Moodley, and I. Varzinczak. First steps in the computation of root justifications. In *Proc. of ARCOE-10*, 2010.
18. N. F. Noy, N. H. Shah, P. L. Whetzel, B. Dai, M. Dorf, N. Griffith, C. Jonquet, D. L. Rubin, M.-A. Storey, C. G. Chute, and M. A. Musen. Biportal: ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Research*, 37:W170–W173, 2009.
19. A. M. Orme, H. Yao, and L. H. Eitzkorn. Complexity metrics for ontology based information. *Int. J. of Technology Management*, 47(1/2/3):161–173, 2009.
20. B. Parsia, E. Sirin, and A. Kalyanpur. Debugging OWL ontologies. In *Proc. of WWW-05*, pages 633–640, 2005.
21. R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–95, 1987.
22. S. Schlobach. Diagnosing terminologies. In *Proc. of AAAI-05*, 2005.
23. S. Schlobach and R. Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In *Proc. of IJCAI-03*, pages 355–362, 2003.
24. B. Suntisrivaraporn, G. Qi, Q. Ji, and P. Haase. A modularization-based approach to finding all justifications for OWL DL entailments. In *Proc. of ASWC-08*, pages 1–15, 2008.
25. S. Tartir, I. Arpinar, M. Moore, A. Sheth, and B. Aleman-Meza. OntoQA: Metric-based ontology quality analysis. In *Proc. of KADASH-05*, 2005.
26. D. Vrandečić and Y. Sure. How to design better ontology metrics. *Proc. of ESWC-07*, 2007.
27. H. Yao, A. Orme, and L. Eitzkorn. Cohesion metrics for ontology design and application. *J. of Computer Science*, 1(1):107–113, 2005.