The Logical Diversity of Explanations in OWL Ontologies

Samantha Bail,
Bijan Parsia,
Ulrike Sattler

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

ABSTRACT

Given the high expressivity of the Web Ontology Language OWL 2, there is a potential for great diversity in the logical content of OWL ontologies. The fact that many naturally occurring entailments of such ontologies have multiple justifications indicates that ontologies often overdetermine their consequences, suggesting a diversity in supporting reasons. On closer inspection, however, we often find that justifications—even for multiple entailments—appear to be structurally similar, suggesting that their multiplicity might be due to diverse *material*, not *formal* grounds for an entailment.

In this paper, we introduce and explore several equivalence relations over justifications for entailments of OWL ontologies which partition a set of justifications into structurally similar subsets. These equivalence relations range from strict isomorphism to looser notions of similarity, covering justifications which contain different class expressions, or even different numbers of axioms. We present the results of a survey of 78 ontologies from the biomedical domain which shows that OWL ontologies used in practice often contain large numbers of structurally similar justifications. We find that a large justification corpus can be reduced by 97% of its original size to a small core of frequently occurring justification templates.

Categories and Subject Descriptors

H.3.5 [On-Line Information Services]: Web-Based Services; I.2.4 [Knowledge Representation Formalisms and Methods]: Semantic Networks; J.3 [Life and Medical Sciences]: Biology and Genetics, Medical Information Systems

Keywords

Semantic Web; OWL; ontologies; explanation; debugging; bioinformatics

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s). Copyright is held by the author/owner(s).

CIKM'13, Oct. 27–Nov. 1, 2013, San Francisco, CA, USA.
ACM 978-1-4503-2263-8/13/10..

1. INTRODUCTION

Since it became a W3C standard in 2004, the Web Ontology Language OWL¹ has become a widely used language for representing knowledge from a variety of domains. OWL is based on the expressive description logic *SRQIQ* [10], which allows the use of complex constructors to model information about the relationships between classes, properties, and individuals. Due to its foundations in description logics and the availability of highly optimised description logic *reasoners*, OWL ontology modellers and users can apply automated reasoning techniques to elicit not only explicitly asserted knowledge in the ontology, but also implicit information which is *entailed* by the ontology. OWL ontologies can be subject to frequent revisions, may be edited collaboratively, or merged with existing ontologies. These processes, paired with the potentially large size and complexity of an ontology, can lead to *errors* which manifest either as logical errors (incoherence and inconsistency) or unwanted entailments which are considered to be incorrect in the context of the domain knowledge modelled in the ontology.

Justifications [12], minimal entailing subsets of an OWL ontology, provide helpful and easy-to-understand explanation support when repairing unwanted entailments in the ontology debugging process. They are currently the prevalent form of explanation in OWL ontology editors such as Protégé 4. While previous research has focused on the issue of making individual justifications easier to understand (e.g. [8, 11, 13]), only little attention has been paid to the occurrence of *multiple justifications* other than the computational problems they often imply. An entailment of an OWL ontology can have a large number of justifications (potentially exponential in the number of axioms in the ontology), with up to several hundreds found in large real-life ontologies [5].

When encountering multiple justifications for a finite set of entailments of an ontology (e.g. unwanted atomic subsumptions, or unsatisfiable classes), we are often faced with a seemingly large and diverse body of reasons why the entailments hold. On closer inspection, however, we frequently find that sets of justifications are very similar, and often even contain structurally identical axioms, with only class, property, and individual names diverging. Pointing out these similarities, and grouping justifications based on their shared structures might greatly assist a user in coping with multiple justifications: rather than trying to understand each individual material justification, the user can focus on understanding the formal *template* of a particular subset of

¹<http://www.w3.org/TR/owl2-overview/>

justifications. Potentially, a user might have to deal with far fewer justifications, thus having a significantly reduced effort when repairing an ontology. This raises several questions: first, how do we determine whether two justifications are structurally similar, and second, how prevalent are such similarities in ontologies used in practice?

A well-known syntactical equivalence relation in OWL is *structural equivalence*. The OWL Structural Specification² states the condition for two OWL objects (named classes, properties, or instances, complex expressions, or OWL axioms) to be structurally equivalent: in short, it defines the objects to be equivalent if they contain the same names and constructors, regardless of ordering and repetition (in an unordered association). The OWL API implements this notion of structural equivalence by default.

A second equivalence relation is *justification isomorphism*³ which was first introduced in a study of the cognitive complexity of justifications [7]: two justifications are isomorphic if they are structurally identical,⁴ i.e. the axioms contain the same *types of* class expressions and only differ in the class, property and instance names. The following example shows two justifications $\mathcal{J}_1$ and $\mathcal{J}_2$ which we consider to be isomorphic:

EXAMPLE 1.

$$\begin{aligned}\mathcal{J}_1 &= \{A \sqsubseteq B \sqcap \exists r.C, B \sqcap \exists r.C \sqsubseteq D\} \models A \sqsubseteq D \\ \mathcal{J}_2 &= \{E \sqsubseteq B \sqcap \exists s.F, B \sqcap \exists s.F \sqsubseteq D\} \models E \sqsubseteq D\end{aligned}$$

It is straightforward to see that in the above justifications the class A in $\mathcal{J}_1$ can be mapped to E in $\mathcal{J}_2$, the property r to s , and the class C to F in order to obtain identical justifications.

While justification isomorphism helps to eliminate the effects of diverging class, property, and instance names, we can also identify types of justifications which may be considered to be very similar despite their use of different constructors:

EXAMPLE 2.

$$\begin{aligned}\mathcal{J}_1 &= \{A \sqsubseteq B \sqcap C, B \sqcap C \sqsubseteq D\} \models A \sqsubseteq D \\ \mathcal{J}_2 &= \{A \sqsubseteq \exists r.C, \exists r.C \sqsubseteq D\} \models A \sqsubseteq D\end{aligned}$$

In this example, the semantics of the complex class expressions $B \sqcap C$ in $\mathcal{J}_1$ and $\exists r.C$ in $\mathcal{J}_2$ are not relevant for the respective entailment; their occurrences in the justifications and their entailments can be replaced by freshly generated class names without affecting the entailment relation. Such a substitution, in turn, would make the two justifications isomorphic. In other words, the complex expressions in $\mathcal{J}_2$ are used in a *propositional* way.

Likewise, justifications of different lengths may be considered similar if their general structure of reasoning is identical:

²<http://www.w3.org/TR/owl2-syntax>

³Note that, in the spirit of consistent naming, we will use the term ‘isomorphism’ for the newly introduced equivalent relations despite their not being isomorphisms (i.e. bijective mappings) in the true sense.

⁴Modulo structural equivalence.

EXAMPLE 3.

$$\begin{aligned}\mathcal{J}_1 &= \{A \sqsubseteq B, B \sqsubseteq C\} \models A \sqsubseteq C \\ \mathcal{J}_2 &= \{A \sqsubseteq B, B \sqsubseteq C, C \sqsubseteq D\} \models A \sqsubseteq D\end{aligned}$$

These two justifications clearly require the same form of reasoning from a user which means we may consider them to be structurally similar in some way. Strict isomorphism only applies to justifications which contain the same number of axioms; it does not cover situations like one shown above. However, for the purpose of structuring sets of justifications and analysing the logical diversity of a corpus of justifications, capturing those kinds of similarities illustrated in the above examples would be highly desirable.

The idea of studying justification patterns, or *rules*, has been explored by Nguyen et al. [15, 14]. In a survey of over 500 OWL ontologies, the authors extracted justification subsets, so-called rules, of up to 4 axioms, and identified the 57 most frequently occurring patterns. These rules were then translated into natural language as used as the basis for proof-like explanations. While this approach (if extended to complete justifications) would capture similar justifications of the type shown in Example 1, it does not cover justifications which differ in the subexpressions or numbers of axioms they use.

In this paper, we introduce two new equivalence relations, *subexpression-isomorphism* $\approx_s$ and *lemma-isomorphism* $\approx_\ell$, between justifications based on the subexpressions of their axioms and axiom subsets, respectively, and show how these equivalence relations can be used to determine the logical diversity of a set of justifications. We then present a detailed survey of ontologies from the biomedical domain in which we investigate the effects of the different types of isomorphism, as well as the effects superfluous expressions have on structural similarity. We find that a surprisingly large number of justifications are indeed structurally similar, which shows us that debugging strategies based on isomorphism templates are indeed applicable to ontologies used in practice.

2. PRELIMINARIES

We assume the reader to be familiar with OWL and the underlying Description Logic *SR_QIQ* [10]. In what follows, $A, B, \dots$ denote class names in an ontology $\mathcal{O}$, r, s role names, and $\text{sig}(\alpha)$ denotes the set of entity names in an OWL axiom α .

Justifications [17, 12] are a type of explanation for entailments of OWL ontologies; they are used in OWL ontology editors such as Protégé 4. A justification is defined as a minimal subset of an ontology $\mathcal{O}$ that causes an entailment η to hold:

DEFINITION 1. $\mathcal{J}$ 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 call such a justification a *self-justification*, and an entailment which has only a self-justification and no other justification in $\mathcal{O}$ a *self-supporting* entailment.

While justifications are minimal with respect to the number of axioms they contain, the axioms themselves may still contain superfluous expressions. A *laconic* justification [8] is a justification which does not contain any superfluous expressions, and for each ‘regular’ justification we can com-

pute a preferred laconic version. The following example illustrates several aspects of non-laconicity in a justification:

EXAMPLE 4.

$$\mathcal{J} = \{A \sqsubseteq B \sqcap \leq 2r.C, \exists r.C \sqsubseteq D\} \models A \sqsubseteq D$$

First, we can reduce the intersection in the first axiom to the single expression $\leq 2r.C$ without affecting the entailment. Second, this expression can then be weakened to $\leq 1r.C$. Third, the class name C in both axioms can be substituted with $\top$. This results in the laconic justification $\{A \sqsubseteq \leq 1r.\top, \exists r.\top \sqsubseteq D\}$.

It is important to note that a justification is always defined with respect to an entailment η . In the remainder of this paper we will therefore use the term *justification* to describe a justification-entailment pair $(\mathcal{J}, \eta)$ where $\mathcal{J}$ is a minimal entailing axiom set for η .

2.1 Justification isomorphism

Justification isomorphism was first introduced as a way to reduce sampling bias in a user study to validate a model for the cognitive complexity of OWL justifications [7]. The study assumed that users would perceive justifications which are structurally identical as equally difficult to understand, taking into account some variation caused by the complexity of class names and domain knowledge. In order to reduce the risk of bias caused by presenting users a series of structurally identical justifications, the study reduced the corpus of 64,800 to 11,600 structurally distinct, i.e. non-isomorphic, justifications. Justification isomorphism is defined as follows:

DEFINITION 2. *Two justifications $(\mathcal{J}_1, \eta_1)$, $(\mathcal{J}_2, \eta_2)$ are isomorphic $((\mathcal{J}_1, \eta_1) \approx_i (\mathcal{J}_2, \eta_2))$ if there exists a bijective renaming ϕ which maps class, property, and instance names in $\mathcal{J}_1$ and η_1 to class, property, and instance names in $\mathcal{J}_2$ and η_2 , respectively, such that $\phi(\mathcal{J}_1) = \mathcal{J}_2$ and $\phi(\eta_1) = \eta_2$.*

Remarks.

1. The relation $\approx_i$ is symmetric, reflexive and transitive, from which it follows that $\approx_i$ is an equivalence relation and thus partitions a set of justifications. Proofs are omitted as these properties are straightforward to see.
2. Justification isomorphism preserves the numbers and types of axioms and subexpressions in the justifications:
 - (a) If $\mathcal{J}_1 \approx_i \mathcal{J}_2$ then $|\mathcal{J}_1| = |\mathcal{J}_2|$.
 - (b) Since the mapping ϕ is bijective, we also have $\mathcal{J}_1 \approx_i \mathcal{J}_2$ implies that $|\text{sig}(\mathcal{J}_1)| = |\text{sig}(\mathcal{J}_2)|$.
 - (c) The sets of logical constructs used in $\mathcal{J}_1$ and $\mathcal{J}_2$ coincide.

3. SUBEXPRESSION-ISOMORPHISM

From the above definition of isomorphism it follows that only justifications with the same number and types of axioms and subexpressions can be isomorphic. It is easy to see, however, that justifications can have a similar structure despite their use of different expressions, as demonstrated in Example 2. Furthermore, obfuscating complex expressions which can be substituted by propositional variables reduces superfluous clutter in a justification, thus improving the readability of a justification akin to laconic justifications. This motivates a notion of *subexpression-isomorphism*

(s-isomorphism), an equivalence relation which allows not only the mapping of class names, but also that of complex subexpressions.

The idea of finding similarities between concepts in description logics has been widely explored in the work on *unification* and *matching*, e.g. [3, 1, 2], for the purpose of detecting redundant class descriptions in ontologies. Given two class expressions C and D , a *unification problem* $C \equiv? D$ asks whether there exists a *substitution* σ , that is, a mapping from a set of named classes in C into the set of class expressions in D , such that $\sigma(C) \equiv \sigma(D)$. The aim of unification is to find a suitable substitution σ which maps atomic classes C to (possibly non-atomic) classes in D such that the two classes are made equivalent.

While the idea behind s-isomorphism is based on unification and matching, the concepts are not directly applicable to the given problem of matching justifications. In our case, the inputs are of different shape from the matching problem: the goal is to unify two sets of axioms and the corresponding entailments, rather than matching a single class expression (a class *pattern*) which contains variables to a class expression. That is, we attempt to identify pairs of possibly complex class expressions that, when replaced with names, result in isomorphic justifications. This means that the substitution mechanism as used in unification is not applicable.

We therefore introduce a *justification template* Θ , which functions as the *unifying justification*, and two substitutions ϕ_1, ϕ_2 for use in s-isomorphism. We will use freshly introduced entity names x_i for the named classes, properties, and instances in a template Θ . In order for s-isomorphism to be transitive, the justifications will have to fulfil certain side-conditions, which will be used in the proof of Proposition 2:

- S1** For any C in the range of ϕ_1 (ϕ_2), C is not equivalent to $\top$.
- S2** For any C in the range of ϕ_1 (ϕ_2) is satisfiable in a *single* element of some interpretation, that is, there is some I such that $|C^I| = 1$.
- S3** For any C_1, C_2 in the range of ϕ_1^{ac} (ϕ_2^{ac} , respectively), it must hold that $\text{sig}(C_1) \cap \text{sig}(C_2) = \emptyset$ and $\text{sig}(C_i) \cap \text{sig}(\Theta) = \emptyset$, that is, the expressions in the domain and the range of the mappings must be pairwise signature disjoint.

DEFINITION 3. *Two justifications $(\mathcal{J}_1, \eta_1)$, $(\mathcal{J}_2, \eta_2)$ are s-isomorphic $((\mathcal{J}_1, \eta_1) \approx_s (\mathcal{J}_2, \eta_2))$ if there exists a justification template (Θ, η) and two injective substitutions ϕ_1, ϕ_2 satisfying **S1** to **S3**, such that*

1. $\Theta \models \eta$
2. $\phi_1(\Theta) = \mathcal{J}_1$ and $\phi_2(\Theta) = \mathcal{J}_2$
3. $\phi_1(\eta) = \eta_1$ and $\phi_2(\eta) = \eta_2$.

The mappings ϕ_1 and ϕ_2 map class, property, and individual names in the template (Θ, η) to subexpressions of $(\mathcal{J}_1, \eta_1)$ and $(\mathcal{J}_2, \eta_2)$, respectively.

Note that in order to be s-isomorphic, the justifications may differ in the number of subexpressions. As with strict isomorphism, however, they must have the same number of axioms: $\mathcal{J}_1 \approx_s \mathcal{J}_2 \rightarrow |\mathcal{J}_1| = |\mathcal{J}_2|$.

PROPOSITION 1. *S-isomorphism is a more general case of strict isomorphism: $\mathcal{J}_1 \approx_i \mathcal{J}_2 \rightarrow \mathcal{J}_1 \approx_s \mathcal{J}_2$.*

PROOF. Let $(\mathcal{J}_1, \eta_1) \approx_i (\mathcal{J}_2, \eta_2)$ via a mapping ϕ . Then we can consider a template Θ to correspond to $\mathcal{J}_1$, the mapping ϕ_1 is the identity relation *id* which maps $\mathcal{J}_1$ to itself, and ϕ_2 corresponds to the mapping ϕ used for strict isomorphism. Thus $\mathcal{J}_1 \approx_i \mathcal{J}_2$ then $\mathcal{J}_1 \approx_s \mathcal{J}_2$. $\square$

PROPOSITION 2. *The relation $\approx_s$ is reflexive, transitive and symmetric for description logics without nominals up to SRIQ .*

PROOF. Reflexivity and symmetry of $\approx_s$ are straightforward to see; therefore we will only outline a proof of the transitivity of s-isomorphism for logics not containing nominals. For a complete detailed proof as well as an example of the non-transitivity of s-isomorphism in the presence of nominals the reader is referred to [4].

Let $(\mathcal{J}_a, \eta_a) \approx_s (\mathcal{J}_b, \eta_b)$ via $\phi_1^{ab}, \phi_2^{ab}, (\Theta_{ab}, \eta_{ab})$ and $(\mathcal{J}_b, \eta_b) \approx_s (\mathcal{J}_c, \eta_c)$ via $\phi_1^{bc}, \phi_2^{bc}, (\Theta_{bc}, \eta_{bc})$, respectively. We aim to show that given this situation, it is always possible to construct two mappings ϕ_1^{ac}, ϕ_2^{ac} and a template (Θ_{ac}, η_{ac}) , such that $(\mathcal{J}_a, \eta_a) \approx_s (\mathcal{J}_c, \eta_c)$ via ϕ_1^{ac}, ϕ_2^{ac} and (Θ_{ac}, η_{ac}) , as illustrated in Figure 1.

We first describe the construction of the (Θ_{ac}, η_{ac}) by creating the parse trees of the three justifications and ‘re-using’ the expressions occurring at the corresponding position in the parse trees of the templates $(\Theta_{ab}, \eta_{ab}), (\Theta_{bc}, \eta_{bc})$. That is, we construct the template (Θ_{ac}, η_{ac}) as a combination of the two existing templates.

The mappings ϕ_1^{ac} and ϕ_2^{ac} are then simply constructed as the mappings from the expressions at the positions in the parse tree of (Θ_{ac}, η_{ac}) to the expressions at their corresponding positions in the parse trees of $(\mathcal{J}_a, \eta_a)$ and $(\mathcal{J}_c, \eta_c)$, respectively.

Finally, we show that the entailment relation $\Theta_{ac} \models \eta_{ac}$ holds by describing the steps to extend any model I_{ac} for Θ_{ac} to a model of η_{ac} by constructing a model I_a of $\mathcal{J}_a$ (or I_c of $\mathcal{J}_c$, respectively) using the side conditions **S1–S3** described above. By showing that it is always possible to construct a template (Θ_{ac}, η_{ac}) for the justifications $\mathcal{J}_a$ and $\mathcal{J}_c$ such that $\mathcal{J}_a \approx_s \mathcal{J}_c$, we have proved the transitivity of the relation $\approx_s$. $\square$

Finally, note that the substitutions ϕ_1 and ϕ_2 are injective, but not surjective. As we have shown in the proof of 2, injectivity of the mappings is a condition for the transitivity of s-isomorphism. However, the mappings are not surjective, as the set of subexpressions in the justifications $\mathcal{J}_1$ and $\mathcal{J}_2$ can be of higher cardinality than the set of class names in Θ (unless the justifications themselves contain no complex expressions). This is illustrated by the following counterexample of non-surjective mappings:

EXAMPLE 5.

$$\begin{aligned} \mathcal{J}_1 &= \{A \sqsubseteq B, B \sqsubseteq C\} \\ \mathcal{J}_2 &= \{A \sqsubseteq B \sqcap \exists r.C, B \sqcap \exists r.C \sqsubseteq D\} \\ \Theta &= \{x_1 \sqsubseteq x_2, x_2 \sqsubseteq x_3\} \\ \phi_1 &= \{x_1 \mapsto A, x_2 \mapsto B, x_3 \mapsto C\} \\ \phi_2 &= \{x_1 \mapsto A, x_2 \mapsto B \sqcap \exists r.C, x_3 \mapsto D\} \end{aligned}$$

The set of *all* subexpressions of $\mathcal{J}_2$ is $\{A, B, \exists r.C, B \sqcap \exists r.C, D\}$. ϕ_2 maps only to a subset of these subexpressions; there exists no mapping for the ‘smaller’ expressions

Figure 1: Graph representing the relations between three justifications which are s-isomorphic via transitivity.

B and $\exists r.C$. This shows that a substitution ϕ can be non-surjective.

Despite the mappings being non-surjective, we require each subexpression to either have a corresponding variable in Θ which maps to it directly, or to occur as part of a larger subexpression which has a corresponding variable in $\mathcal{J}$. This guarantees that the isomorphic justifications have the same number of subexpressions which correspond to variables in $\mathcal{J}$.

4. LEMMA-ISOMORPHISM

While s-isomorphism covers a number of justifications that can be regarded as equivalent due to them requiring the same type of reasoning to reach the entailment, it only applies to justifications which have the same number of axioms. S-isomorphism does not take into account cases where the justifications differ only marginally in some subset, but where the general reasoning may be regarded as similar nonetheless. We therefore introduce the notion of *lemma-isomorphism* (l-isomorphism), which extends s-isomorphism with the substitution of subsets of justifications through intermediate entailments, so-called *lemmas*. The general motivation behind l-isomorphism is demonstrated by the following example:

EXAMPLE 6.

$$\begin{aligned} \mathcal{J}_1 &= \{A \sqsubseteq B, B \sqsubseteq C\} \models A \sqsubseteq C \\ \mathcal{J}_2 &= \{A \sqsubseteq B, B \sqsubseteq C, C \sqsubseteq D\} \models A \sqsubseteq D \end{aligned}$$

It is straightforward to see that both $\mathcal{J}_1$ and $\mathcal{J}_2$ require the same type of reasoning from a human user. As the justifications only differ in the length of the atomic subsumption chains that lead to the entailment, we can certainly consider them to be *similar* with respect to *some* similarity measure. However, the two justifications are not considered isomorphic with respect to the definitions for strict isomorphism or s-isomorphism. We therefore introduce a new type of isomorphism which takes into account the fact that subsets of justifications can be replaced with intermediate entailments which follow from them.

A lemma λ for a justification $\mathcal{J}$ is an entailment of a subset S of $\mathcal{J}$. In order to define l-isomorphism, we need to introduce two more concepts: *tidy* axiom sets, and justification *lemmatisations*, that is, justifications which are *enriched* with lemmas, as defined by Horridge [6].

Tidy axiom sets prevent meaningless lemmatisations by ensuring that the set S that generates a lemma is consistent and does not entail synonyms for $\top$ or $\perp$. They are defined as follows [6]:

DEFINITION 4. A set of axioms S is tidy if $S \not\models \top \sqsubseteq \perp$, $S \not\models A \sqsubseteq \perp$ and $S \not\models \top \sqsubseteq A$ for all $A \in \text{sig}(S)$.

The following definition of *summarising lemmatisations* is based on the definition of lemmatisations given by Horridge [6], however, with some modifications: first, Horridge's definition considers the *cognitive complexity* of a lemmatisation in order to ensure that a lemmatisation is *easier* to understand than the justification it is based on. As the complexity of a lemmatisation is not relevant for l-isomorphism, we drop this condition. Second, the original definition is too weak for use in l-isomorphism, as we will show with an example below; thus, we introduce a new condition for the justification subset S which ensures that the lemmas used for the lemmatisation are *summarising*. And finally, we simplify the definition to omit some unneeded details:

DEFINITION 5. Let $(\mathcal{J}, \eta)$ be a justification, let $S = \bigcup S_i$ be a set of tidy subsets of $\mathcal{J}$, and let $\Lambda_S = \bigcup \lambda_i$ be a set of lemmas such that $S_i \models \lambda_i$. The set $\mathcal{J}^{\Lambda_S} = (\mathcal{J} \setminus S) \cup \Lambda_S$ is called a summarising lemmatisation of $\mathcal{J}$ if

1. $\mathcal{J}^{\Lambda_S}$ is a justification for η ,
2. S_i are pairwise disjoint.

If clear from the context, a summarising lemmatisation $\mathcal{J}^{\Lambda_S}$ may also be called a *lemmatised justification*.

PROPOSITION 3. Given a $S_i \in S$ that is substituted with a lemma λ_i in a summarising lemmatisation $\mathcal{J}^{\Lambda_S}$, there is no set $S' \subset S_i$ such that $S' \models \lambda_i$.

PROOF. This proposition follows from the minimality condition for justifications: given a justification $(\mathcal{J}, \eta)$, we remove a subset $S_i \subseteq \mathcal{J}$ and replace it with a lemma λ_i such that $(\mathcal{J} \setminus S_i) \cup \{\lambda_i\}$ is a justification for η . If there exists a subset $S' \subset S_i$ such that $S' \models \lambda_i$, then it also holds that $(\mathcal{J} \setminus S_i) \cup S' \models \eta$. Therefore, the axioms $S_i \setminus S'$ are redundant with respect to the entailment η , which violates the minimality condition for $\mathcal{J}^{\Lambda_S}$. $\square$

Given the definitions for lemmatisations, we can now define lemma-isomorphism as an extension to subexpression-isomorphism:

DEFINITION 6. Two justifications $(\mathcal{J}_1, \eta_1)$, $(\mathcal{J}_2, \eta_2)$ are lemma-isomorphic $((\mathcal{J}_1, \eta) \approx_\ell (\mathcal{J}_2, \eta))$ if there exist summarising lemmatisations $\mathcal{J}_1^{\Lambda_{S1}}$, $\mathcal{J}_2^{\Lambda_{S2}}$ which are s-isomorphic: $\mathcal{J}_1^{\Lambda_{S1}} \approx_s \mathcal{J}_2^{\Lambda_{S2}}$.

4.1 Restrictions on lemmatisations

L-isomorphism using *arbitrary* summarising lemmatisations as defined above carries two undesirable properties: first, the lemmatised justifications might differ strongly from the original justifications; in the most extreme case, the lemmatisation of a justification can be the entailment itself. We therefore have to introduce an additional constraint that ensures that a lemmatisation is *understandable* to a user. Second, we cannot guarantee that l-isomorphism using arbitrary lemmas is in fact transitive, which means that l-isomorphism would lose the desirable property of being an equivalence relation. In order to demonstrate l-isomorphism in this paper, we therefore restrict lemmatisations to a very simple pattern, namely *maximal atomic subsumption chains*.

Maximal atomic subsumption chains.

While research on human understanding of description logic is fairly limited, an exploratory study by Horridge et al. [7] found that chains of atomic subsumption axioms of the type $S = \{A_0 \sqsubseteq A_1, A_1 \sqsubseteq A_2 \dots A_{n-1} \sqsubseteq A_n\}$ are one of the easiest recognizable patterns for OWL ontology users. Given a chain S in a justification, often only the relation between the subclass A_0 in the first axiom and the superclass A_n in the last axiom is relevant for understanding how the axioms in S contribute to the justification. That is, the subsumption chain produces a summarising lemma $\lambda = A_0 \sqsubseteq A_n$.

We call an atomic subsumption chain $S = A_0 \sqsubseteq \dots \sqsubseteq A_n$ *maximal* if there is no longer chain $S' = A_0 \sqsubseteq \dots \sqsubseteq A_{n+k}$ for $k \geq 1$. It is possible for a set of axioms to form an *apparently* maximal subsumption chain in which only a subset of the chain can be substituted by a summarising lemma, as shown in the following example:

EXAMPLE 7.

$$\begin{aligned} \mathcal{J} = \{ & A \sqsubseteq \exists r.C, \\ & A \sqsubseteq B, \\ & B \sqsubseteq C, \\ & C \sqsubseteq D, \\ & C \sqcap \exists r.D \sqsubseteq E \} \models A \sqsubseteq E \end{aligned}$$

The axiom set $S' = \{A \sqsubseteq B, B \sqsubseteq C, C \sqsubseteq D\}$ in Example 7 is a maximal atomic subsumption chain in $\mathcal{J}$ which entails $A \sqsubseteq D$. It is clear to see, however, that the entailment does not hold in the lemmatisation $\mathcal{J} \setminus S \cup \{A \sqsubseteq D\}$ as the important lemmas $A \sqsubseteq C$ and $C \sqsubseteq D$ would be lost; that is, the lemmatisation based on S' is not summarising. We can only substitute the non-maximal chain $S = \{A \sqsubseteq B, B \sqsubseteq C\}$ with its entailment $A \sqsubseteq C$ in order to obtain a summarising lemmatisation.

While substituting non-maximal chains is not problematic in itself, an implementation of the atomic subsumption chain lemmatisation to detect non-maximal chains would require searching through all subchains of the maximal atomic subsumption chain in order to find a suitable, substitutable, chain. Therefore, our current implementation only attempts to substitute maximal chains, which may lead to false negative results when attempting to compare two justifications which are only l-isomorphic through lemmatisations using non-maximal chains.

Non-transitivity.

We can show [4] that allowing *arbitrary* summarising lemmatisations can lead to non-transitivity of l-isomorphism for justifications which contain *multiple* reasons for an entailment to hold, a phenomenon called *internal masking* [9]. Thus, disallowing justifications with internal masking would eliminate one cause of non-transitivity. However, whether this is the only restriction we have to apply in order to preserve the transitivity of lemma-isomorphism remains an open question. For the time being, we treat all results of the l-isomorphism implementation as possible approximations. This means that it is possible that some sets of justifications may be considered to be non-isomorphic when they actually are. Hence, we may potentially over-estimate the logical diversity of a corpus of justifications and *under-estimate* the effects of l-isomorphism, which we consider acceptable for the purpose of demonstrating the concept of l-isomorphism in this paper.

5. BIOPORTAL ONTOLOGY SURVEY

In this section, we present a survey of a corpus of OWL ontologies from the biomedical domain. The purpose of this survey is to analyse the structural similarities of justifications which are representative, in terms of size and expressivity, for the range of ontologies used in practise. In order to prevent bias through hand-picking ‘suitable’ ontologies, our aim was to use an independently motivated corpus of ontologies. Thus, the choice was between a random sample of web ontologies (e.g. obtained from a web crawl) and an existing ontology repository. As the NCBO BioPortal ontology repository [16] contains a large number of actively used and well-studied ontologies which cover a broad spectrum of size and expressivity, it seemed a suitable choice for an extensive survey of justificatory structure.

5.1 Data generation

Ontology corpus.

The NCBO BioPortal is a web-based ontology repository which provides over 300 ontologies published by research groups from the biomedical domain. It also includes the full set of OBO Foundry⁵ ontologies, which use a flat-file format that can be translated into OWL 2; therefore, the OBO ontologies were included in the corpus. At the time of downloading (May 2012), 322 OWL and OBO files were listed in BioPortal, of which 256 could be downloaded via the BioPortal REST interface and successfully parsed by the OWL API. Each downloadable ontology was merged with its imports closure and serialised as OWL/XML file. For practical reasons, a processing timeout of 20 minutes was set for all reasoning tasks (consistency checking and classification). The overall timeout per ontology for each experiment was set to 90 minutes. The reasoners used were JFact version 0.9 and Pellet version 2.3.0. All experiments were run on a Mac Mini with a 2.7 GHz Intel Core i7 processor, with 16 GB RAM assigned to the JVM.

Justification computation.

Due to the small number of ‘naturally occurring’ unsatisfiable classes in published OWL ontologies, we shifted our analysis from obvious ‘errors’, that is, unsatisfiable classes, to atomic subsumptions which represent the class hierarchy of an ontology. We computed the set of atomic subsumptions of type $A \sqsubseteq B$ for named, satisfiable classes A and B of each ontology, including self-supporting entailments, and excluding tautologies such as $A \sqsubseteq \top$. We then computed all justifications for the entailments and filtered out all ‘trivial’ justifications, i.e. self-justifications as well as justifications which consisted only of an atomic subsumption chain. Due to performance issues, the justification generation and isomorphism detection was limited as follows:

- The justification generation was restricted to a random sample of 1,000 entailments per ontology and 500 justifications per entailment.
- Justifications containing more than 10 axioms or conjunctions/disjunctions with more than 5 operands were excluded from this part of the study.

The resulting set S_s consisted of non-trivial 141,560 justifications for 19,097 entailments from 78 ontologies in the corpus.

⁵<http://www.obofoundry.org/>

Table 1: Basic ontology metrics in the corpus.

	Mean	Median	Min	Max
Classes	1,887	34	4	38,640
Object properties	38	24	0	431
Data properties	9	2	0	178
Individuals	56	0	0	1,697
Logical axioms	3,871	931	5	68,777
Justifications	1,828	756	1	25,765

Ontology properties.

The 78 ontologies in the corpus span a broad spectrum of sizes and expressivities, ranging from ontologies with as few as 4 classes and 5 logical axioms to as many as 38,640 classes and 68,777 axioms. Some of the basic metrics for the ontologies in the corpus are shown in Tables 1. A complete list of the 78 ontologies (i.e. those containing some non-trivial justifications) including their metrics and experiment results can be found online.⁶

Regarding their expressivity, 64 ontologies are OWL 2 DL ontologies, of which 18 are in the OWL 2 EL profile, 8 in OWL 2 QL, and 7 in OWL 2 RL. The remaining 14 ontologies were OWL 2 Full. Note that the three OWL 2 profiles are *not* exclusive, that is, an ontology can fall into several profiles. The description logic complexity in the corpus ranges from $\mathcal{EL}^{++}$ (the 18 ontologies in the OWL 2 EL profile) to full OWL 2 DL expressivity (*SRQIQ*, 4 ontologies), with several ontologies being in a highly expressive description logic (*SHQ* or *SRQ* including either inverse properties *I* or nominals *O*).

5.2 Implementation

The general idea behind the implementation of a program to check whether two justifications are isomorphic is the comparison of the *parse trees* of the two justifications. Each axiom in a justification is parsed into its parse tree where the nodes are labelled with either entity names (properties or classes), OWL constructors, or integers (in the case of cardinality restrictions). The parse tree of a justification simply contains each axiom parse tree as child of a blank root node. The algorithm then traverses the trees and, at each node, performs a comparison to check whether the nodes can be matched according to the rules of the given isomorphism. While details of the implementation are omitted for space reasons, the complete code is made available online.⁷

5.3 Results

In order to determine the frequency of isomorphic justifications in the corpus, we analysed the justifications for the entailments in S_s in three different experiments:

Ex. 1: justifications for a *single* entailment.

Ex. 2: justifications for *all* entailments within an ontology.

Ex. 3: justifications for *all* entailments of *all* ontologies in the corpus.

The isomorphism statistics were generated for all three types of isomorphism, that is, strict, subexpression-, and lemma-isomorphism.

The mean times per ontology required to determine isomorphism between justifications (including parsing the jus-

⁶<https://sites.google.com/site/isocikm2013/>

⁷<https://sites.google.com/site/isocikm2013/>

tifications and existing templates from file) ranged between 22.8 seconds (for strict isomorphism between justifications for individual entailments) and 192.3 seconds (for checking s-isomorphism within an ontology).

5.3.1 Ex. 1: Individual entailments

The first experiment investigates the occurrence of structurally similar justifications for the justifications of each individual entailment in the corpus, which gives us an indication of the applicability of isomorphism-based debugging techniques for individual entailments with multiple justifications.

Strict isomorphism.

Strict isomorphism reduces the average number of justifications per entailment from 7.4 to 4.9 templates per entailment ($\sigma = 9.5$, $m = 2$), which is a reduction by 33.7% compared to the full corpus.

On average, a template covers 1.5 justifications (standard deviation $\sigma = 2.3$, median $m = 1$), with some ontologies containing entailments with large numbers of isomorphic justifications. One such example is the *Orphanet Ontology of Rare Diseases*, whose dominating templates are of the type

$$\Theta_1 = \{C1 \sqsubseteq C2, C2 \sqsubseteq \exists p1.C4, \text{domain}(p1, C3)\} \models C1 \sqsubseteq C3$$

with atomic subsumption chains of arbitrary size in place of the first subsumption axiom, and some variations that include subproperty axioms. Two of the templates of this type cover the majority (110 and 105 justifications, respectively) of the 220 justifications each for several entailments in the ontology. From personal contact with the *Orphanet* developers we have learned that this OWL ontology is in fact generated automatically from an existing medical database, which explains the frequent occurrences of uniform justifications.

Subexpression-isomorphism.

S-isomorphism across the justifications of individual entailments only affects a very small number of entailments in the corpus. The average number of templates per entailment, compared to the full corpus, remains nearly identical at 4.915 for strict and 4.903 for s-isomorphism ($\sigma = 9.5$, $m = 2$). Compared to strict isomorphism, the number of templates is reduced by only 0.3%. This small reduction does not affect the average number of justifications per template, which remains at 1.5 justifications. Overall, only 166 entailments (0.9% of the total corpus) from 15 of the 78 ontologies are affected by s-isomorphism; the remaining 63 ontologies contain the same number of templates as for strict isomorphism.

Lemma-isomorphism.

While s-isomorphism does not have a strong effect on individual entailments, l-isomorphism shows a more visible reduction in template numbers.⁸ On average, the justifications are reduced to 4.7 templates per entailment ($\sigma = 8.8$, $m = 2$), which is reduction by 4.1% compared to both strict and s-isomorphism.

⁸Note that S_s does not contain any justifications that consist entirely of atomic subsumption chains. That is, all atomic subsumption chains that are affected by l-isomorphism will be strict subsets of the justifications in the corpus.

Table 2: Reduction and template coverage per ontology.

Type	Mean reduction	Mean coverage
strict	73.1%	8.2
subex	74%	8.8
lemma	78.2%	12

A total of 1,492 entailments (7.8% of the total corpus) from 43 ontologies are affected by l-isomorphism, with an average reduction of 30.3% compared to strict isomorphism for those entailments. The strongest effects can be seen in the *Fission Yeast Phenotype* ontology, where the justifications for several entailments only differ in the length of their atomic subsumption chains and thus are each reduced to a single template of the type

$$\Theta_2 = \{C1 \sqsubseteq \dots \sqsubseteq C_n, C_n \equiv C2 \sqcap \dots\} \models C1 \sqsubseteq C2.$$

5.3.2 Ex. 2: Isomorphism within ontologies

Across the justifications for all entailments of an ontology, the reductions caused by the three equivalence relations are clearly more visible than for individual entailments. However, the effects of the relations differ strongly across the 78 ontologies, with strict isomorphism generally having the strongest impact, and s-isomorphism having the lowest impact. Overall, only three ontologies show no effect for *any* of the three types of isomorphism; two of these contain only a single justification (which means there is no reduction possible), while one ontology (*OBOE SBC*) contains 5 distinct justifications. Table 2 shows an overview of the mean reduction and template coverage for the three isomorphism types.

Strict isomorphism.

Compared to the results for individual entailments, the effects of the equivalence relations are much more significant if we consider the justifications for *all* entailments in an ontology. On average, an ontology in S_s contains 1,814.9 justifications; these are reduced to 436.1 templates through strict isomorphism, which is an average reduction by 73.1%.

4 ontologies with small numbers of justifications (less than 10) are not affected by strict isomorphism, while the remaining 74 ontologies show reductions of up to 99.6%. 28 of those ontologies are reduced by more than 90%; with an average number of 3,092 justifications per ontology, these reductions reveal significant numbers of structurally similar justifications.

The templates cover an average of 8.2 justifications each ($\sigma = 22.4$, $m = 2$), with some of the highest coverage occurring in the *Orphanet* ontology. The 3,948 justifications in this ontology can be reduced to only 14 templates, which are all variations (featuring atomic subsumption chains of varying length) of the template Θ_2 we have shown above.

Subexpression-Isomorphism.

S-isomorphism reduces the justifications in the corpus by an average of 74% per ontology, which is only a small change compared to strict isomorphism. The number of justifications covered by a single template is slightly increased to 8.8 ($\sigma = 23.5$, $m = 2$) justifications per template.

Again, the majority of ontologies (44 of 78) are not affected by s-isomorphism, whereas 5 ontologies show reduc-

Table 3: Template frequency and coverage (mean, median, min, max) across the corpus.

Type	Count	% of S_s	Coverage			
			Mean	Med	Min	Max
all	141,560	100%	-	-	-	-
strict	12,527	8.8%	11.3	2	1	2,072
subex	10,952	7.8%	12.9	2	1	2,128
lemma	5,487	3.1%	25.8	3	1	7,490

tions of between 20% and 38.3% compared to strict isomorphism. This includes the *Bleeding History Phenotype* ontology (1,158 justifications, 60 isomorphism templates, 37 s-isomorphism templates), which contains a number of justifications of the type

$$\{C1 \sqsubseteq \exists p1.(C2 \sqcup C3), \text{domain}(p1, C4)\} \models C1 \sqsubseteq C4$$

which is s-isomorphic to justifications that contain a named class in place of the disjunction $C2 \sqcup C3$ in the subsumption axiom, thus matching template Θ_1 .

Closer inspection of the *Lipid* ontology (3,258 justifications, 1,762 isomorphism templates, 1,471 s-isomorphism templates) reveals that a large number of justifications in the ontology consist of a single equivalence axiom of the form $C1 \sqsubseteq C2 \sqcap x$ with the entailment being $C1 \sqsubseteq C2$. The remainder of the conjunction, represented by x , consists of a number of complex expressions of varying length and nesting depth. While s-isomorphism captures these types of justifications (since the remainders x can all be matched against each other), the actual reason for their similarity lies in their identical *cores* $C1 \sqsubseteq C2$, with the remainder x being a superfluous part.

Lemma-isomorphism.

As for single entailments, l-isomorphism has a stronger effect on the justification corpus than s-isomorphism when applied across each ontology, as it reduces the justifications in an ontology by an average of 78.2%.

A template covers an average of 12 justifications ($\sigma = 38$, $m = 3$) in each ontology, which is a visible increase from the 8.8 justifications covered by s-isomorphism. 13 ontologies (with an average number of 32.1 justifications) show no reaction to l-isomorphism, whereas 12 ontologies containing large numbers of justifications (mean = 1,555.8) see a reduction of 41.7% and more, up to 82.1% compared to s-isomorphism.

5.3.3 Ex. 3: Isomorphism across the corpus

In the final stage of our analysis of isomorphism in the BioPortal corpus, we will look at the templates spanning all justifications for all entailments across the ontologies in the corpus. While we have seen that isomorphic justifications occur frequently within an ontology, we now analyse the structural similarity of justifications across multiple ontologies. Table 3 gives an overview of the numbers of templates in S_s for the three isomorphism types against the full justification corpus, as well as the coverage (number of justifications) per template.

Strict isomorphism.

The 141,560 justifications in S_s are reduced to only 12,527 templates, which is a reduction by 91.2%. On average, 11.3 justifications ($\sigma = 54.3$, $m = 2$) share the same template, with the 8 most frequent templates covering over 1,000 justifications each. The most frequent templates (by numbers of justifications covered) are all variations of the template Θ_1 , that is, a combination of a domain axiom and a subsumption axiom with an existential restriction on the RHS, including some additional subsumption axioms, an equivalence axiom in place of a subsumption, or variations of the filler.

If we look at the spread of templates across the ontologies in the corpus, we find that only 8.7% of the templates cover justifications in multiple ontologies, with a maximum spread of 26 ontologies for two templates which are, again, variations of Θ_1 . Figure 2a shows the frequency distributions of the justification templates across the justifications and ontologies in S_s . The y-position of a data point indicates the number of justifications a template covers is, and the bubble size indicates the number of ontologies a template occurs in.

We can see that a small number of templates covers large numbers of justifications, with a steep drop to 10 and less justifications per template around the 2,000 mark. If we take a closer look at the numbers, we find that the majority of justifications in the corpus (81.2%) are covered by the 2,000 most frequent templates (out of 12,527), and a third (33.1%) of justifications are even covered by the 100 most frequent templates.

Subexpression-isomorphism.

The effects of s-isomorphism across the corpus are only marginal compared to strict isomorphism. The justifications are reduced to 10,952 templates, which is an overall reduction by 92.2% and a 1% difference compared to strict isomorphism. The number of justifications covered by a single template is slightly increased with an average of 12.9 justifications ($\sigma = 59$, $m = 2$) per template.

The most frequent template (by numbers of justifications) is again Θ_2 , which covers 2,128 (1.5% of the total set) justifications in 26 ontologies. Across the ontologies in the corpus, the most frequent template occurs in 28 of the 78 ontologies. This template is a single equivalence axiom which we have already seen in the *Lipid* ontology:

$$\Theta_3 = \{C1 \equiv C2 \sqcap x\} \models C1 \sqsubseteq C2$$

The superfluous part x matches a number of operands such as atomic classes and existential restrictions. Interestingly, while this template occurs in the highest number of ontologies, it only covers 573 justifications across the corpus.

Lemma-isomorphism.

Across all justifications in the corpus, l-isomorphism has a clearly visible impact. The 141,560 justifications are reduced to only 5,487 templates, which is less than half as many templates as the strictly isomorphic ones, and an overall reduction of 96.9%. Figure 2b shows the frequency of templates for l-isomorphism.

On average, a template covers 25.8 justifications ($\sigma = 208.5$, $m = 3$); however, the large standard deviation shows that the distribution of justifications per template has shifted towards a few very frequent templates, whereas there is still a ‘long tail’ of 1,878 templates that match only a single justification. If we consider the distribution of justifications per template over the quartiles of the corpus, 25% of the

(a) Template frequency for strict isomorphism.

(b) Template frequency for l-isomorphism.

Figure 2: Template frequencies across the corpus for strict and lemma-isomorphism.

justifications in S_s can be covered by the 8 most frequent templates, 50% by the 44 most frequent templates, and 75% by the 277 (out of 5,487) most frequent templates. The most frequent templates are all subtle variations of a template containing only two or three axioms:

$$\begin{aligned}\Theta_4 &= \{C1 \sqsubseteq C2, C3 \equiv C2 \sqcup C4, C3 \sqsubseteq C5\} \models C1 \sqsubseteq C5 \\ \Theta_5 &= \{C1 \sqsubseteq C2, C5 \equiv C2 \sqcup C3\} \models C1 \sqsubseteq C5 \\ \Theta_6 &= \{C1 \sqsubseteq C2, C3 \equiv C2 \sqcup C4 \sqcup C6, C3 \sqsubseteq C5\} \models C1 \sqsubseteq C6 \\ \Theta_7 &= \{C1 \sqsubseteq C2, C5 \equiv C2 \sqcup C3 \sqcup C4\} \models C1 \sqsubseteq C5\end{aligned}$$

Note that any subsumption axiom in the templates Θ_4 to Θ_7 corresponds to an atomic subsumption chain of arbitrary length. These templates each cover between 4,206 and 7,490 justifications, which amounts to 17% of all justifications in the corpus. If we look at the number of ontologies a template occurs in, the most frequent templates are Θ_3 and Θ_5 , both of which can be found in 28 of the 78 ontologies in the corpus. Overall, however, as with strict and s-isomorphism, only a fraction of the templates (8.5%) occur in multiple ontologies, while the majority of templates (5,018 out of 5,487) can only be found in a single ontology.

5.3.4 Isomorphism and superfluity

As we have already seen in our discussion of s-isomorphism, a justification can potentially be non-isomorphic only due to it containing superfluous subexpressions that do not contribute to the entailment. In order to determine to which extent superfluous parts affect isomorphism, we computed the laconic versions for justifications in S_s , which resulted in a corpus S_{sl} containing 47,667 laconic justifications, of which 31.6% were now atomic subsumption chain justifications or self-justifications. The number of laconic justifications is significantly lower compared to S_s for two reasons: first, due to the high runtime of the laconic justification generation mechanism, some justifications could not be computed; hence, due to timeouts, the number of entailments in S_{sl} is only 18,300 (compared to 19,097 in S_s). More striking, however, is the effect of removing superfluous expressions on the equality between justifications: multiple regular justifications for individual entailments frequently resulted in *the same* laconic justification, which drastically reduces the

number of justifications despite the difference in the number of entailments being only small. Table 4 shows a comparison of the cross-corpus reductions for laconic and regular justifications in S_s and S_{sl} , respectively.

Across S_{sl} , strict isomorphism reduces the justifications to 3,653 templates, which is a reduction to 7.6% (compared with 8.8% for the regular justifications in S_s). 8 out of the 10 most frequent templates (by number of justifications they cover) are atomic subsumption chains between 1 and 8 axioms. Interestingly, this also indicates that, despite the apparent complexity of many justifications in the corpus (recall that we only computed the laconic versions of *non-trivial* justifications), the actual reasoning behind those justifications comes down to simple atomic subsumption.

Perhaps unsurprisingly, s-isomorphism only has a small effect on S_{sl} . The laconic justifications are reduced to 2,936 templates, which is a reduction to 6.2% of the corpus. Just as with strict isomorphism, this is only a marginally stronger reduction compared to the corpus of regular justifications (7.8%).

Finally, l-isomorphism reduces the laconic justifications in S_{sl} to only 1,789 templates, which is an overall reduction to 3.8% of the set of laconic justifications. In terms of the overall proportion this is a smaller reduction than for l-isomorphism in regular justifications (3.1%). However, since the set S_{sl} is only roughly a quarter of the size of S_s , this smaller *relative* reduction is not too surprising. As we could already expect based on the prevalent templates we found for strict isomorphism, the most frequent template is a single atomic subsumption axiom, which covers atomic subsumption chain justifications of arbitrary length. This template covers the 15,066 justifications (31.6%) in the corpus we have already mentioned above, and can be found in 61 of the 78 ontologies.

Table 4: Comparison of reductions in S_s and S_{sl} .

Type	S_s (regular)		S_{sl} (laconic)	
	Count	% of S_s	Count	% of S_{sl}
all	141,560	100%	46,667	100%
strict	12,527	8.8%	3,653	7.8%
subex	10,952	7.8%	2,036	6.2%
lemma	5,487	3.1%	1,789	3.8%

6. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced two new equivalence relations between justifications for entailments of OWL ontologies, subexpression-isomorphism $\approx_s$ and lemma-isomorphism $\approx_\ell$, which capture various degrees of structural similarities between justifications. Our analysis of justification isomorphism in the BioPortal corpus revealed that strict isomorphism and lemma-isomorphism have a significant impact on the landscape of justifications in our sample: the 141,560 justifications in the corpus are effectively reduced to only 5,487 distinct templates. If we remove all superfluous expressions from the justification axioms, this logical similarities between justifications are even more visible, as the justifications can be reduced to only 1,789 templates, which is just over 1% of the size of the original corpus. This analysis shows us that the logical diversity of OWL justifications can be considerably lower than it appears, while the small effect of s-isomorphism also indicates that there exists a certain amount of *real* logical diversity in the modelling of ontologies, and that complex subexpressions are generally *not* used in a propositional way.

For application in OWL debugging tools, both strict isomorphism and lemma-isomorphism seem promising as the basis for improved debugging techniques, such as grouping structurally similar justifications and presenting a user with an abstract template rather than a set of justifications. As we have seen, this can significantly reduce the number of justifications a user has to inspect in order to understand the reasoning behind them, thus reducing their effort when it comes to repairing the justifications. For future work, we are planning to design and implement such a debugging tool, for example as a plugin to the Protégé 4 editor, and conduct user evaluation studies to determine the usefulness of such a tool.

7. REFERENCES

- [1] F. Baader, R. Küsters, A. Borgida, and D. L. McGuinness. Matching in description logics. *J. of Logic and Computation*, 9(3):411–447, 1999.
- [2] F. Baader and B. Morawska. Unification in the description logic $\mathcal{EL}$. In *Proc. of RTA-09*, pages 350–364, 2009.
- [3] F. Baader and P. Narendran. Unification of concept terms in description logics. In *Proc. of ECAI-98*, pages 331–335, 1998.
- [4] S. Bail. *The Justificatory Structure of OWL Ontologies*. PhD thesis, The University of Manchester, 2013.
- [5] S. Bail, M. Horridge, B. Parsia, and U. Sattler. The justificatory structure of the NCBO BioPortal ontologies. In *Proc. of ISWC-11*, 2011.
- [6] M. Horridge. *Justification Based Explanation in Ontologies*. PhD thesis, The University of Manchester, 2011.
- [7] M. Horridge, S. Bail, B. Parsia, and U. Sattler. The cognitive complexity of OWL justifications. In *Proc. of ISWC-11*, 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. Justification masking in OWL. In *Proc. of DL-10*, 2010.
- [10] I. Horrocks, O. Kutz, and U. Sattler. The even more irresistible *SROIQ*. In *Proc. of KR-06*, 2006.
- [11] A. Kalyanpur, B. Parsia, and B. Cuenca Grau. Beyond asserted axioms: Fine-grain justifications for OWL-DL entailments. In *Proc. of DL-06*, 2006.
- [12] A. Kalyanpur, B. Parsia, E. Sirin, and J. Hendler. Debugging unsatisfiable classes in OWL ontologies. *J. of Web Semantics*, 3(4):268–293, 2005.
- [13] J. S. C. Lam, J. Z. Pan, D. Sleeman, and W. Vasconcelos. A fine-grained approach to resolving unsatisfiable ontologies. In *Proc. of WI-06*. Springer, 2006.
- [14] T. A. T. Nguyen, R. Power, P. Piwek, and S. Williams. Measuring the understandability of deduction rules for OWL. In *Proc. of WoDOOM-12*, 2012.
- [15] T. A. T. Nguyen, R. Power, P. Piwek, and S. Williams. Planning accessible explanations for entailments in OWL ontologies. In *Proc. of INLG 2012*, pages 110–114, 2012.
- [16] 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. Bioportal: ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Research*, 37:W170–W173, 2009.
- [17] 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.