Justification Patterns for OWL DL Ontologies

Other

How to cite:

Nguyen, Tu Anh T.; Power, Richard; Piwek, Paul and Williams, Sandra (2011). Justification Patterns for OWL DL Ontologies. Department of Computing, The Open University.

Version: Version of Record

Link(s) to article on publisher's website:
http://computing-reports.open.ac.uk/2011/TR2011-05.pdf

ISSN 1744-1986

Technical Report N o 2011/ 05

Justification Patterns for OWL DL Ontologies

Tu Anh T. Nguyen and Richard Power and Paul Piwek and Sandra Williams

25 May, 2011

Department of Computing
Faculty of Mathematics, Computing and Technology
The Open University

Walton Hall, Milton Keynes, MK7 6AA
United Kingdom

http://computing.open.ac.uk

Justification Patterns for OWL DL Ontologies

Tu Anh T. Nguyen, Richard Power, Paul Piwek, Sandra Williams

Department of Computing, The Open University, UK
`{t.nguyen,, r.power,, p.piwek,, s.h.williams}@open.ac.uk`

Abstract. For debugging OWL-DL ontologies, natural language explanations of inconsistencies and undesirable entailments are of great help. From such explanations, ontology developers can learn why an ontology gives rise to specific entailments. Unfortunately, commonly used tableaux-based reasoning services do not provide a basis for such explanations, since they rely on a refutation proof strategy and normalising transformations that are difficult for human ontology editors to understand. For this reason, we investigate the use of automatically generated justifications for entailments (i.e., minimal sets of axioms from the ontology that cause entailments to hold). We show that such justifications fall into a manageable number of patterns, which can be used as a basis for generating natural language explanations by associating each justification pattern with a rhetorical pattern in natural language.

Keywords: OWL, Justifications, Natural Language Explanation

1 Introduction

Since being endorsed as a standard language for encoding ontologies in 2004, OWL (Web Ontology Language) has been used increasingly in developing significant domain ontologies such as the gene ontology[1] and the SNOMED CT ontology[2]. Building such ontologies requires a significant amount of effort and expertise. To simplify the development task, graphical editing tools have been developed [9, 6] linked to tableaux-based reasoners [12, 10] which compute implicit statements (i.e. entailments) and inconsistencies that follow from axioms in the ontology. When inconsistencies occur or entailments appear to be undesirable, ontology developers need to understand them in order to debug the ontology, so some kind of explanation would be helpful. However, tableaux-based reasoning services do not provide a basis for generating explanations, because they depend on a refutation proof strategy and normalising transformations which are unintuitive to humans [2]. For example, it is unnatural to explain the entailment $C \sqsubseteq \exists R.D$ (i.e. every individual in class C must have at least one R relation to an individual in class D) with reference to the subsumption $(C \sqcap \forall R.\neg D) \sqsubseteq \emptyset$.

Recently several research groups have explored ways of computing justifications for entailments and inconsistencies in OWL-DL ontologies [8, 11, 5, 4]. A

---

[1] See `http://www.geneontology.org/`

[2] See `http://www.ihtsdo.org/snomed-ct/`

justification for an entailment is a minimal set of axioms, drawn from the ontology, that is sufficient for the entailment to arise [7]. For logicians, a justification is a helpful explanation of an entailment as it pinpoints precisely the axioms that cause the entailment to hold [7]. Human domain experts, however, require further explanation of the meaning of each axiom and the inference process linking axioms to the entailment. Let us give an example.

Table 1 shows a sample justification for the entailment '*milli* is a *PrefixOrUnit*' from the units[3] ontology, which contains about 500 axioms. The justification has four axioms, presented here in OWL Functional Syntax. To help human domain experts, we might provide an explanation such as the following:

> *milli* is a *PrefixOrUnit* because (a) *milli* is *Prefix*, and (b) every *Prefix* is a *PrefixOrUnit*. Statement (a) can be inferred from axioms 1, 3 and 4; in words, *milli* is a *Prefix* because we know from axioms 1 and 4 that *milli_meter* has as prefix only a *Prefix*, and from axiom 3 that *milli_meter* has as prefix *milli*. Statement (b) can be inferred from axiom 2.

The main aim of this paper is to show that automatically generated justifications do provide a good basis for generating explanations such as the one given above. The main problem we address is that the number of different justifications is in theory unlimited. This, at first sight, prevents development of a manageable set of rhetorical patterns in natural language for expressing justifications. We will, however, show that given certain abstractions, the justifications from a large collection of OWL-DL ontologies do belong to limited set of patterns.

We proceed as follows. In Section 2, we present the formulation of two types of patterns for justifications, specific and abstract. Next, in Section 3, we describe an empirical study of justification patterns from a corpus of 191 OWL-DL ontologies. Section 4 discusses the results of the study, and Section 5 concludes.

Table 1. Justification for the entailment *ClassAssertion(PrefixOrUnit,milli)* in OWL Functional Syntax

No	Axiom
1	*ClassAssertion(UnitDerivedByScaling,milli_meter)*
2	*EquivalentClasses(PrefixOrUnit,ObjectUnionOf(Prefix,Unit))*
3	*ObjectPropertyAssertion(hasPrefix,milli_meter,milli)*
4	*SubClassOf(UnitDerivedByScaling,ObjectAllValuesFrom(hasPrefix,Prefix))*

2 From Justifications to Justification Patterns

Justifications for entailments of OWL-DL ontologies can be very diverse, for four reasons. First, although the size of a justification is usually small compared with

---

[3] See `http://sweet.jpl.nasa.gov/ontology/units.owl`

that of an ontology, it may still contain dozens of axioms: an empirical experiment on a corpus of OWL-DL ontologies shows variations from 1 up to 40 [8]. Secondly, justifications may contain many different types of axioms; currently, OWL-DL allows about 30 axiom functors and 17 class constructors — and these class constructors can be nested indefinitely. Thirdly, even when justifications use the same axiom types and class constructors, their argument distribution may differ. The first two justifications in Table 2, for example, have similar size and structure, but different argument arrangements (2-3-1 in the first, 3-1-2 in the second). Finally, since axioms in justifications are taken without modification from those asserted by ontology builders, they might include superfluous parts. For instance, in the justification 'd' in Table 2, the expression *ObjectSomeValuesFrom* in axiom 3 makes no contribution to the entailment; we only need to infer from this axiom that $ConductionFibres \sqsubseteq Myocardium$.

Table 2. Justifications of the abstract pattern *CLASS-INCLUSION-CHAIN*

No	Entailment	Justification
a	*WaterBody* $\sqsubseteq$ *PlanetaryStructure*	1.*EquivalentClasses(PlanetaryStructure,EarthRealm)* 2.*EquivalentClasses(WaterBody,BodyOfWater)* 3.*SubClassOf(BodyOfWater,EarthRealm)*
b	*DrySeasonDuration* $\sqsubseteq$ *Occurrence*	1.*EquivalentClasses(Event,Duration)* 2.*EquivalentClasses(Occurrence,Event)* 3.*SubClassOf(DrySeasonDuration,Duration)*
c	*LongWaveRadiation* $\sqsubseteq$ *ElectromagWave*	1.*EquivalentClasses(ElectromagRadiation,ElectromagWave)* 2.*EquivalentClasses(InfraredRadiation,LongWaveRadiation)* 3.*SubClassOf(InfraredRadiation,ElectromagRadiation)*
d	*PurkinjeFibres* $\sqsubseteq$ *CardiacMuscle*	1.*EquivalentClasses(CardiacMuscle,Myocardium)* 2.*EquivalentClasses(ConductionFibres,PurkinjeFibres)* 3.*EquivalentClasses(ConductionFibres,ObjectIntersectionOf(Myocardium, ObjectSomeValuesFrom(hasSpecificFunction,Conduction))*

Given the diversity of justifications, there might be some doubt as to whether we can find a generic set of rules for mapping them to rhetorical patterns in natural language. Fortunately, the empirical study we report here shows that in practice most justifications conform to a smaller number of common patterns, some patterns more frequent than others. Moreover, many patterns contain other patterns within them, which leads to the possibility of representing these patterns in a more abstract way. To investigate these patterns, we analyse justifications at two levels of abstraction: specific and abstract. Specific patterns for justifications retain all axioms and class functors, but abstract over atomic terms; this is done by substituting names of entities (i.e. classes, individuals, properties, . . .) by alpha-numeric identifiers. Specifically, names of atomic classes are replaced by capital letters A, B, . . . , those of individuals by *i0*, *i1*, . . . , those of object properties by *r0*, *r1*, . . . , and those of data properties by *d0*, *d1*, . . . Specific patterns of all justifications in Table 2 are presented in Table 3; since patterns a and c are exactly the same, they can be amalgamated.

Table 3. Specific patterns of justifications in Table 2

No	Entailment	Justification
a, c	*SubClassOf(A,B)*	1.*EquivalentClasses(A,C)* 2.*EquivalentClasses(B,D)* 3.*SubClassOf(C,D)*
b	*SubClassOf(A,B)*	1.*EquivalentClasses(B,D)* 2.*EquivalentClasses(C,D)* 3.*SubClassOf(A,C)*
d	*SubClassOf(A,B)*	1.*EquivalentClasses(A,C)* 2.*EquivalentClasses(B,D)* 3.*EquivalentClasses(C,ObjectIntersectionOf(D,ObjectSomeValuesFrom(r0,E))*

Abstract patterns are based on a deeper abstraction over groups of axioms which have certain inference steps in common. They have been derived by manual analysis of specific patterns. For example, although the first two specific patterns in Table 3 are different, they both infer the subsumption $A \sqsubseteq B$ through a chain of class inclusions (i.e. a chain of *EquivalentClasses* and *SubClassOf* relations) on named classes: *A-C-D-B*. Therefore, they can both be represented by an abstract pattern called *CLASS-INCLUSION-CHAIN(A,B)*, which represents a chain of class inclusions from named class A to named class B with no limitation on the number of intermediate named classes used in the chain.

In the last specific pattern in Table 3, if we can get the subsumption $C \sqsubseteq D$ from the last axiom, the whole pattern will also belong to the abstract pattern *CLASS-INCLUSION-CHAIN(A,B)*. Therefore, it would be useful if we can apply basic transformations on axioms of specific patterns before analysing abstract patterns to remove superfluous parts. Several basic transformations that can be applied to axioms of specific patterns are presented in Table 4.

Following the above-mentioned procedure we can formulate two other abstract patterns, namely *OBJECT-PROPERTY-DOMAIN(r0,A)* and *OBJECT-PROPERTY-RANGE(r0,A)*. Each pattern represents a set of specific patterns from which we can infer that class A is the domain or range of the property *r0*. Specific patterns for *OBJECT-PROPERTY-DOMAIN(r0,A)* and *OBJECT-PROPERTY-RANGE(r0,A)* collected from our data are shown in Table 5.

3 Method

We have analysed justification patterns in a large corpus of OWL-DL ontologies, comprising 191 ontologies of varying size, subject and expressivity taken from the TONES Ontology Repository [1]. This repository is a database of OWL-DL ontologies designed for use by developers, especially in testing automated

Table 4. Basic transformations on axioms of specific patterns. *<ClassExp>* is either a named class or a complex class expression.

No	Input Axiom	Inferred Axiom
1	$A \equiv \langle ClassExp1 \rangle \sqcap \langle ClassExp2 \rangle \sqcap \ldots$	$+\ A \sqsubseteq \langle ClassExp1 \rangle$ $+\ A \sqsubseteq \langle ClassExp2 \rangle$ $+\ \ldots$
2	$A \sqsubseteq \langle ClassExp1 \rangle \sqcap \langle ClassExp2 \rangle \sqcap \ldots$	$+\ A \sqsubseteq \langle ClassExp1 \rangle$ $+\ A \sqsubseteq \langle ClassExp2 \rangle$ $+\ \ldots$
3	$A \equiv \langle ClassExp1 \rangle \sqcup \langle ClassExp2 \rangle \sqcup \ldots$	$+\ \langle ClassExp1 \rangle \sqsubseteq A$ $+\ \langle ClassExp2 \rangle \sqsubseteq A$ $+\ \ldots$

Table 5. Specific patterns for abstract patterns *OBJECT-PROPERTY-DOMAIN(r0,A)* and *OBJECT-PROPERTY-RANGE(r0,A)*

No	DOMAIN	RANGE
1	*ObjectPropertyDomain(r0,A)*	*ObjectPropertyRange(r0,A)*
2	*EquivalentClasses(A, ObjectSomeValuesFrom(r0,Thing))*	—
3	*InverseObjectProperties(r0,r1)* *ObjectPropertyRange(r1,A)*	*InverseObjectProperties(r0,r1)* *ObjectPropertyDomain(r1,A)*
4	*ObjectPropertyDomain(r1,A)* *SubObjectPropertyOf(r0,r1)*	*ObjectPropertyRange(r1,A)* *SubObjectPropertyOf(r0,r1)*
5	*InverseObjectProperties(r0,r1)* *ObjectPropertyRange(r2,A)* *SubObjectPropertyOf(r1,r2)*	*InverseObjectProperties(r0,r1)* *ObjectPropertyDomain(r2,A)* *SubObjectPropertyOf(r1,r2)*
6	*ObjectPropertyDomain(r1,A)* *SubObjectPropertyOf(r0,r2)* *SubObjectPropertyOf(r2,r1*	*ObjectPropertyRange(r1,A)* *SubObjectPropertyOf(r0,r2)* *SubObjectPropertyOf(r2,r1)*
7	*ObjectPropertyRange(r1,A)* *SubObjectPropertyOf(r0,r1)* *SymmetricObjectProperty(r1)*	— — —

reasoning techniques. Currently, the repository consists of 218 ontologies from a wide range of sources; however, 27 are either inconsistent, faulty, or incompatible with both the Pellet and FaCT++ reasoners. Since at this stage we focus only on explaining justifications for possibly undesirable entailments in consistent ontologies, we ignore these ontologies here. The size of ontologies in our corpus ranges from 1 to 233,582 axioms. We have developed a Java-based program to compute and analyse justification patterns, relying on the Pellet and FaCT++ reasoners for computing entailments, and on a program developed by Kalyanpur and Horridge [8] for computing justifications for each entailment. In addition, the OWL API package [3] was employed to parse the structure of axioms.

Justifications can be distinguished by the nature of their entailments. Currently, we focus on just three types of entailment, namely class assertion (i.e., *ClassAssertion(<Class>,<Individual>)*), class subsumption between two named classes (i.e., *SubClassOf(<Class>,<Class>)*), and class unsatisfiability (i.e. *SubClassOf(<Class>,Nothing)*). For each ontology in the corpus, our program begins by computing all entailments of these types; then for each entailment we compute up to 10 different justifications. (We decided to compute at most 10 justifications for each entailment, partly to reduce processing time, and partly because most entailments have fewer than 10 justifications.)

To automatically compute specific patterns for justifications, justifications were first converted to OWL Functional Syntax. Then, within each justification, we replaced every occurrence of an entity URI by an alpha-numeric identifier drawn from a small standard set, as illustrated in the previous section. Sorting algorithms were then performed both at axiom and argument levels, before and after every substitution, to make sure that similar justifications would have the same specific patterns. Justifications having common specific patterns were grouped, and their frequencies calculated by two different measures: (a) occurrences of the justification pattern across all ontologies (called ‘justification frequency’), and (b) the number of ontologies in which the pattern occurred at least once (called ‘ontology frequency’). Ontology frequency is a more stable measure since it is relatively unaffected by differences in ontology size — a pattern could occur in very few ontologies, but have a high justification frequency because these ontologies were very large. These two measures of frequency were also used for abstract patterns, which were tabulated manually from the data for specific patterns.

4 Results

Table 6 shows the overall results of our study. Among 191 ontologies in the corpus, only 108 ontologies have at least one non-empty justification[4]. We collected over 76,000 non-empty justifications, more than 90 percent of which contain seven or fewer axioms. There are approximately 2,800 specific patterns that all

---

[4] For a top-level class A that was not explicitly described as a sub-class of *Thing*, an entailment $A \sqsubseteq Thing$ would be generated. The justification for this entailment is an empty set. We ignored these justifications.

justifications conform to, which is much smaller than the total number of justifications. However, specific patterns are not strongly focussed. Tables 7, 8 and 9 in the Appendix section show the top ten specific patterns for each entailment type ordered by ontology frequency then justification frequency; as can be seen, these cover at most one-third of all cases. We now consider each entailment type separately.

Table 6. Distribution of justifications and specific patterns

Entailment Type	**Justification Frequency**	**Ontology Frequency**	**Specific Patterns**
ClassAssertion(<Class>,<Individual>)	44875 (58.9%)	41(38.0%)	452 (16.3%)
SubClassOf(<Class>,<Class>)	29211 (38.3%)	106(98.1%)	1660 (59.9%)
SubClassOf(<Class>,Nothing)	2135 (2.8%)	43(39.8%)	659 (23.8%)
TOTAL	76221(100.0%)	—	2771(100.0%)

4.1 Abstract Patterns for Class Assertion Justifications

A justification for a *ClassAssertion(A,i0)* entailment first locates an individual *i0* in some class B, then shows that B is a sub-class of A. If the first step happens to locate *i0* directly in A, then the second step is unnecessary. Similarly, if the second step finds that A is equivalent to class *Thing* then the first step is no longer needed. In general, the second step is obviously reducible to the problem of justifying a subsumption relationship between two classes. Table 10 shows abstract patterns and their corresponding frequencies over the whole data set. Patterns are listed in three groups sorted by ontology frequency: (i) first step only, (ii) second step only, and (iii) both steps.

4.2 Abstract Patterns for Class Subsumption Justifications

A justification for a *SubClassOf(A,B)* entailment can either show that class B is equivalent to class *Thing*, or that A is subsumed by B using chains of class inclusions or many other properties. Table 11 shows abstract patterns and their corresponding frequencies. Patterns are listed in two groups: (i) normal class subsumption patterns and (ii) B is equivalent to *Thing*.

4.3 Abstract Patterns for Class Unsatisfiability Justifications

Abstract patterns and corresponding frequencies for class unsatisfiability justifications can be found in Table 12 in the Appendix.

5 Conclusion

Our study of over 76,000 non-empty justifications from a corpus of 191 ontologies empirically confirms the hypothesis that a manageable number of justification patterns can cover most justifications. In particular, we found that although specific patterns based on the size and structure of justifications are not strongly focussed, we were able to find focussed abstract patterns based on recurring inference steps.

Regarding the problem of generating natural language explanations of justifications for OWL-DL ontologies, the study is helpful in at least three ways:

- it confirms the feasibility of an approach which relies on a relatively small set of justification patterns,
- it provides a set of basic transformations on axioms to reduce superfluous parts in justifications, and
- it identifies significant inference steps that should be helpful in automatically constructing rhetorical patterns for explaining justifications in natural language.

References

1. The TONES Ontology Repository. `http://owl.cs.manchester.ac.uk/repository/`, Accessed: 30th August 2010
2. Borgida, A., Franconi, E., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F.: Explaining $\mathcal{ALC}$ Subsumption. In: DL 1999, International Workshop on Description Logics. Linköping, Sweden (1999)
3. Horridge, M., Bechhofer, S.: The OWL API: A Java API for Working with OWL 2 Ontologies. In: OWLED 2009, 6th OWL Experienced and Directions Workshop. Virginia, USA (2009)
4. Horridge, M., Parsia, B., Sattler, U.: Explaining Inconsistencies in OWL Ontologies. In: SUM 2009, 3rd International Conference on Scalable Uncertainty Management. pp. 124–137. Washington D.C., USA (2009)
5. Ji, Q., Qi, G., Haase, P.: A Relevance-Directed Algorithm for Finding Justifications of DL Entailments. In: ASWC 2009, 4th Asian Semantic Web Conference. pp. 306–320. Berlin, Heidelberg (2009)
6. Kaljurand, K., Fuchs, N.: Verbalizing OWL in Attempto Controlled English. In: OWLED 2007, 3rd OWL Experienced and Directions Workshop. Innsbruck, Austria (2007)
7. Kalyanpur, A.: Debugging and repair of OWL ontologies. Ph.D. thesis, University of Maryland (2006)
8. Kalyanpur, A., Parsia, B., Horridge, M., Sirin, E.: Finding All Justifications of OWL DL Entailments. In: ISWC 2007, 6th International Semantic Web Conference. Busan, Korea (2007)
9. Knublauch, H., Musen, M.A., Rector, A.L.: Editing Description Logic Ontologies with the Protégé OWL Plugin. In: DL 2004, International Workshop on Description Logics. British Columbia, Canada (2004)
10. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A Practical OWL-DL Reasoner. Journal of Web Semantics 5, 51–53 (2007)

11. Suntisrivaraporn, B., Qi, G., Ji, Q., Haase, P.: A Modularization-Based Approach to Finding All Justifications for OWL DL Entailments . In: ASWC 2008, 3rd Asian Semantic Web Conference. Bangkok, Thailand (2008)
12. Tsarkov, D., Horrocks, I.: FaCT++ Description Logic Reasoner: System Description. In: IJCAR 2006, 3rd International Joint Conference on Automated Reasoning. vol. 4130, pp. 292–297. Seattle, Washington, USA (2006)

Appendix: Tables for Specific and Abstract Patterns

Table 7. Top 10 specific patterns for the entailment *ClassAssertion(A,i0)*. *SubClassOf* axioms are presented in logic notation due to space limitation.

No	Specific Pattern	Ontology Frequency	Justification Frequency
1	*ClassAssertion(B,i0)* $B \sqsubseteq A$	31(75.6%)	2566 (5.7%)
2	*ClassAssertion(B,i0)* $B \sqsubseteq C \sqsubseteq A$	26(63.4%)	2571 (5.7%)
3	*ObjectPropertyAssertion(r0,i0,i1)* *ObjectPropertyDomain(r0,A)*	17(41.5%)	2626 (5.9%)
4	*ObjectPropertyAssertion(r0,i1,i0)* *ObjectPropertyRange(r0,A)*	17(41.5%)	1880 (4.2%)
5	*ClassAssertion(B,i0)* $B \sqsubseteq C \sqsubseteq D \sqsubseteq A$	16(39.0%)	1927 (4.3%)
6	*ObjectPropertyAssertion(r0,i0,i1)* *ObjectPropertyDomain(r0,B)* $B \sqsubseteq A$	14(34.1%)	2513 (5.6%)
7	*ClassAssertion(B,i0)* $B \sqsubseteq C \sqsubseteq D \sqsubseteq E \sqsubseteq A$	12(29.3%)	1324 (3.0%)
8	*ClassAssertion(B,i0)* *EquivalentClasses(A,objectUnionOf(C,D))* $B \sqsubseteq E \sqsubseteq F \sqsubseteq C$	12(29.3%)	265 (0.6%)
9	*ClassAssertion(B,i0)* *EquivalentClasses(A,objectUnionOf(B,C)*	12(29.3%)	222 (0.5%)
10	*ClassAssertion(B,i0)* *EquivalentClasses(A,B)*	12(29.3%)	188 (0.4%)
	Total	—	16082(35.8%)

Table 8. Top 10 specific patterns for the entailment *SubClassOf(A,B)*. *SubClassOf* axioms are presented in logic notation due to space limitation.

No	**Specific Pattern**	**Ontology Frequency**	**Justification Frequency**
1	*EquivalentClasses(B,objectUnionOf(A,C))*	47(44.3%)	440 (1.5%)
2	*EquivalentClasses(A,objectIntersectionOf(* *B,objectSomeValuesFrom(r0,C)))*	31(29.2%)	1553 (5.3%)
3	*EquivalentClasses(B,objectUnionOf(A,C,D))*	31(29.2%)	226 (0.8%)
4	*EquivalentClasses(B,objectUnionOf(A,C,D,E))*	28(26.4%)	238 (0.8%)
5	*ObjectPropertyDomain(r0,B)* *SubClassOf(A,objectSomeValuesFrom(r0,C))*	27(25.5%)	284 (1.0%)
6	*EquivalentClasses(A,C)* $C \sqsubseteq B$	23(21.7%)	2527 (8.7%)
7	$A \sqsubseteq C \sqsubseteq B$	23(21.7%)	457 (1.6%)
8	*EquivalentClasses(B,objectUnionOf(A,C,D,E,F))*	20(18.9%)	204 (0.7%)
9	*EquivalentClasses(B,C)* $A \sqsubseteq C$	18(17.0%)	3663(12.5%)
10	*EquivalentClasses(A,objectIntersectionOf(B,C))*	18(17.0%)	63 (0.2%)
	Total	—	9655(33.1%)

Table 9. Top 10 specific patterns for the entailment *SubClassOf(A,Nothing)*. *SubClassOf* axioms are presented in logic notation due to space limitation.

No	Specific Pattern	Ontology Frequency	Justification Frequency
1	*DisjointClasses(B,C)* $A \sqsubseteq B$ $A \sqsubseteq D \sqsubseteq E \sqsubseteq C$	9(20.9%)	16(0.7%)
2	*DisjointClasses(B,C)* $A \sqsubseteq D \sqsubseteq B$ $A \sqsubseteq E \sqsubseteq C$	8(18.6%)	16(0.7%)
3	*FunctionalDataProperty(d0)* $A \sqsubseteq B \sqsubseteq \ldots \sqsubseteq F$ $A \sqsubseteq G \sqsubseteq \ldots \sqsubseteq J$ *SubClassOf(F,dataHasValue(d0,l0))* *SubClassOf(J,dataHasValue(d0,l1))*	8(18.6%)	8(0.4%)
4	*FunctionalDataProperty(d0)* $A \sqsubseteq B \sqsubseteq \ldots \sqsubseteq F$ $A \sqsubseteq G \sqsubseteq \ldots \sqsubseteq L$ *SubClassOf(F,dataHasValue(d0,l0))* *SubClassOf(L,dataHasValue(d0,l1))*	8(18.6%)	8(0.4%)
5	*FunctionalDataProperty(d0)* $A \sqsubseteq B \sqsubseteq \ldots \sqsubseteq G$ $A \sqsubseteq H \sqsubseteq \ldots \sqsubseteq N$ *SubClassOf(G,dataHasValue(d0,l0))* *SubClassOf(N,dataHasValue(d0,l1))*	8(18.6%)	8(0.4%)
6	*FunctionalDataProperty(d0)* $A \sqsubseteq B \sqsubseteq \ldots \sqsubseteq H$ $A \sqsubseteq I \sqsubseteq \ldots \sqsubseteq N$ *SubClassOf(H,dataHasValue(d0,l0))* *SubClassOf(N,dataHasValue(d0,l1))*	8(18.6%)	8(0.4%)
7	*DisjointClasses(B,C)* $A \sqsubseteq C$ $A \sqsubseteq D \sqsubseteq B$	6(14.0%)	26(1.2%)
8	*DisjointClasses(B,C)* $A \sqsubseteq D \sqsubseteq B$ $A \sqsubseteq E \sqsubseteq F \sqsubseteq C$	6(14.0%)	15(0.7%)
9	*DisjointClasses(B,C)* $A \sqsubseteq D \sqsubseteq E \sqsubseteq B$ $A \sqsubseteq D \sqsubseteq F \sqsubseteq C$	5(11.6%)	25(1.2%)
10	*DisjointClasses(A,B)* $A \sqsubseteq C \sqsubseteq D \sqsubseteq E \sqsubseteq B$	5(11.6%)	13(0.6%)
	Total	—	143(6.7%)

Table 10. Abstract patterns for *ClassAssertion(A,i0)* entailment and corresponding frequencies. Patterns are listed in three groups as illustrated in Section 4.1.

Abstract Pattern	**Ontology Frequency**	**Justification Frequency**
(i) First Step Only		
ObjectPropertyAssertion(r0,i0,i1) *OBJECT-PROPERTY-DOMAIN(r0,A)*	17(41.5%)	2655 (5.9%)
ObjectPropertyAssertion(r0,i1,i0) *OBJECT-PROPERTY-RANGE(r0,A)*	17(41.5%)	1908 (4.3%)
DataPropertyAssertion(d0,i0,l0) *DataPropertyDomain(d0,A)*	10(24.4%)	905 (2.0%)
ClassAssertion(B,i1) *CLASS-INCLUSION-CHAIN(B,C)* *ObjectPropertyRange(r0,A)* *SubClassOf(C,ObjectHasValue(r0,i0))*	9(22.0%)	87 (0.2%)
ClassAssertion(B,i1) *CLASS-INCLUSION-CHAIN(B,C)* *ObjectPropertyAssertion(r0,i1,i0)* *SubClassOf(C,ObjectAllValuesFrom(r0,A))*	8(19.5%)	284 (0.6%)
(ii) Second Step Only		
CLASS-INCLUSION(ObjectAllValuesFrom(r0,...),B) *ObjectPropertyDomain(r0,A)* *CLASS-INCLUSION-CHAIN(B,A)*	1 (2.4%)	1832 (4.1%)
(iii) Both Steps		
ClassAssertion(B,i0) *CLASS-INCLUSION-CHAIN(B,A)*	32(78.0%)	12375 (27.6%)
ObjectPropertyAssertion(r0,i0,i1) *OBJECT-PROPERTY-DOMAIN(r0,B)* *CLASS-INCLUSION-CHAIN(B,A)*	14(34.1%)	9836 (21.9%)
ObjectPropertyAssertion(r0,i1,i0) *OBJECT-PROPERTY-RANGE(r0,B)* *CLASS-INCLUSION-CHAIN(B,A)*	14(34.1%)	1871 (4.2%)
DataPropertyAssertion(d0,i0,l0) *DataPropertyDomain(d0,B)* *CLASS-INCLUSION-CHAIN(B,A)*	11(26.8%)	4187 (9.3%)
ClassAssertion(B,i1) *CLASS-INCLUSION-CHAIN(B,C)* *ObjectPropertyAssertion(r0,i1,i0)* *SubClassOf(C,ObjectAllValuesFrom(r0,D))* *CLASS-INCLUSION-CHAIN(D,A)*	10(24.4%)	2495 (5.6%)
ClassAssertion(B,i1) *CLASS-INCLUSION-CHAIN(B,C)* *ObjectPropertyRange(r0,D)* *SubClassOf(C,objectHasValue(r0,i0))* *CLASS-INCLUSION-CHAIN(D,A)*	9(22.0%)	247 (0.6%)
Other Patterns	—	6193 (13.8%)
TOTAL	—	44875(100.0%)

Table 11. Abstract patterns for *SubClassOf(A,B)* entailment and corresponding frequencies. Patterns are listed in two groups as illustrated in Section 4.2.

Abstract Pattern	**Ontology Frequency**	**Justification Frequency**
(i) Normal Patterns		
CLASS-INCLUSION-CHAIN(A,B)	83(78.3%)	13459 (46.1%)
CLASS-INCLUSION-CHAIN(A,C) *OBJECT-PROPERTY-DOMAIN(r0,D)* *SubClassOf(C,<ObjectPropertyExp>(r0,...))* *CLASS-INCLUSION-CHAIN(D,B)*	45(42.5%)	3246 (11.1%)
CLASS-INCLUSION(B,ObjectIntersectionOf(C,<ObjPropExp>(r0,D))) *CLASS-INCLUSION-CHAIN(A,C)* *CLASS-INCLUSION-CHAIN(A,<ObjPropExp>(r1,E)))* *SubObjectProperty(r1,r0)* *CLASS-INCLUSION-CHAIN(E,D)*	27(25.5%)	1135 (3.9%)
CLASS-INCLUSION-CHAIN(A,C) *DataPropertyDomain(d0,D)* *SubClassOf(C,<DataPropertyExp>(d0,...))* *CLASS-INCLUSION-CHAIN(D,B)*	11(10.4%)	5657 (19.4%)
CLASS-INCLUSION(B,ObjectIntersectionOf(C,<ObjPropExp 1>(r0,D))) *CLASS-INCLUSION-CHAIN(A,C)* *CLASS-INCLUSION-CHAIN(A,<ObjPropExp1>(r0,E)))* *CLASS-INCLUSION-CHAIN(E,<ObjPropExp1>(r0,F)))* *TransitiveObjectProperty(r0)* *CLASS-INCLUSION-CHAIN(F,D)*	5 (4.7%)	405 (1.4%)
DisjointUnion(B,A,...)	1 (0.9%)	118 (0.4%)
(ii) B ≡ Thing		
CLASS-INCLUSION(ObjectAllValuesFrom(r0,...),C) *ObjectPropertyDomain(r0,B)* *CLASS-INCLUSION-CHAIN(C,B)*	3 (2.8%)	186 (0.6%)
Other Patterns	—	5005 (17.1%)
TOTAL	—	29211 (100.0%)

Table 12. Abstract patterns for *SubClassOf(A,Nothing)* entailment and corresponding frequencies.

Abstract Pattern	**Ontology Frequency**	**Justification Frequency**
DisjointClasses(B,C) *CLASS-INCLUSION-CHAIN(A,B)* *CLASS-INCLUSION-CHAIN(A,C)*	14 (32.6%)	594 (27.8%)
DisjointClasses(A,B) *CLASS-INCLUSION-CHAIN(A,B)*	12 (27.9%)	55 (2.6%)
DisjointClasses(B,C) *CLASS-INCLUSION-CHAIN(A,B)* *CLASS-INCLUSION-CHAIN(A,C)* *SubClassOf(B,dataHasValue(d0,l0))* *SubClassOf(C,dataHasValue(d0,l1))* *FunctionalDataProperty(d0)*	9 20.9%)	33 (1.5%)
DataPropertyRange(d0,DR0) *CLASS-INCLUSION-CHAIN(A,. . . B . . .)* *CLASS-INCLUSION-CHAIN(B,* *. . . <DataPropExp>(d0,l1/DR1). . .)* where *l1* is not of the type *DT0* or *DT1* is not *DT0*	3 (7.0%)	40 (1.9%)
DisjointClasses(B,C) *CLASS-INCLUSION-CHAIN(A,C)* *CLASS-INCLUSION(ObjectAllValuesFrom(r0,. . .),D)* *OBJECT-PROPERTY-DOMAIN(r0,B)* *CLASS-INCLUSION-CHAIN(D,B)*	1 (2.3%)	73 (3.4%)
DisjointClasses(B,C) *CLASS-INCLUSION(ObjectAllValuesFrom(r0,. . .),D)* *OBJECT-PROPERTY-DOMAIN(r0,B)* *CLASS-INCLUSION-CHAIN(D,B)* *CLASS-INCLUSION-CHAIN(A,E)* *EquivalentClasses(E,objectSomeValuesFrom(r1,F))* *CLASS-INCLUSION-CHAIN(F,C)*	1 (2.3%)	58 (2.7%)
DisjointClasses(B,C) *CLASS-INCLUSION(ObjectAllValuesFrom(r0,. . .),D)* *OBJECT-PROPERTY-DOMAIN(r0,C)* *CLASS-INCLUSION-CHAIN(D,C)* *CLASS-INCLUSION-CHAIN(A,F)* *EquivalentClasses(F,ObjectSomeValuesFrom(r1,. . .))* *ObjectPropertyRange(r1,E)* *CLASS-INCLUSION-CHAIN(E,B)*	1 (2.3%)	54 (2.5%)
FunctionalObjectProperty(r0) *CLASS-INCLUSION-CHAIN(A,. . . B . . .)* *CLASS-INCLUSION-CHAIN(B,* *. . . ObjectExactCardinality(n,r0,C) . . .)* where n is not equal to 1	1 (2.3%)	8 (0.4%)
DisjointClasses(A,B) *CLASS-INCLUSION(ObjectAllValuesFrom(r0,. . .),C)* *OBJECT-PROPERTY-DOMAIN(r0,B)* *CLASS-INCLUSION-CHAIN(C,B)*	1 (2.3%)	6 (0.3%)
DisjointClasses(A,B) *OBJECT-PROPERTY-DOMAIN(r0,B)* *CLASS-INCLUSION-CHAIN(A,<ObjPropExp>(r0,. . .))*	1 (2.3%)	2 (0.1%)
Other patterns	—	1212 (56.8%)
TOTAL	—	2135 (100.0%)