The OWL Import Benchmark Suite.
Formal description of the benchmarks

Introduction

This document provides a formal description of the ontologies that compose the OWL Import Benchmark Suite[1].
Figure 1 shows a sample of the description: each entry comes with a description in both natural language and in the Description Logics formalism.

ID	Here there is the description in natural language... ...and here the one in the Description Logics formalism.
ISE06	Import a single object property with domain a class and range multiple classes $\top \sqsubseteq \forall hasChild^{-}.\mathsf{Person}$ $\top \sqsubseteq \forall hasChild.\mathsf{Person}$ $\top \sqsubseteq \forall hasChild.\mathsf{Human}$ $\top \sqsubseteq \forall hasChild.\mathsf{Child}$

Table 1: Structure of the tables and a sample instantiation

The formalism presented in this document uses the following conventions [1]:

Axiom	**DL**
Class (C partial $D_1 \ldots D_n$)	$C \sqsubseteq (D_1 \sqcap \ldots \sqcap D_n)$
Class (C complete $D_1 \ldots D_n$)	$C \equiv (D_1 \sqcap \ldots \sqcap D_n)$
DisjointClasses($C_1 \ldots C_n$)	$C_1 \sqsubseteq \neg C_n$
EquivalentClasses($C_1 \ldots C_n$)	$(C_1 \equiv C_n)$
SubClassOf($C_1 C_2$)	$(C_1 \sqsubseteq C_2)$
Property(P	
domain($D_1 \ldots D_n$)	$\top \sqsubseteq \forall P^{-}.D_i;\ \forall 1 \leq i \leq n$
range($D_1 \ldots D_n$)	$\top \sqsubseteq \forall P.D_i;\ \forall 1 \leq i \leq n$
super($Q_1 \ldots Q_n$)	$P \sqsubseteq Q_i;\ \forall 1 \leq i \leq n$
inverseOfQ	$P \equiv Q^{-}$
Symmetric	$P \equiv P^{-}$
Transitive	$P^{+} \sqsubseteq P$
Functional	$\top \sqsubseteq \forall P$
InverseFunctional	$\top \sqsubseteq \forall P^{-}$
)	
SameIndividuals(($o_1 \ldots o_n$))	$(o_1 = o_i);\ \forall 1 \leq i \leq n$
DifferentIndividuals(($D_1 \ldots D_n$))	$\neg(o_i = o_j);\ \forall 1 \leq i \leq j \leq n$

[1] http://knowledgeweb.semanticweb.org/benckmarking_interoperability/owl/import.html

1 Class benchmarks

Group A: Class hierarchies

ISA01	Import a single class Person
ISA02	Import a single class, subclass of a second class which is subclass of a third one $\mathsf{Child} \sqsubseteq \mathsf{Man} \sqsubseteq \mathsf{Person}$
ISA03	Import a class that is subclass of two classes $\mathsf{Child} \sqsubseteq \mathsf{Man}$ $\mathsf{Child} \sqsubseteq \mathsf{Person}$
ISA04	Import several classes subclass of a single class $\mathsf{Woman} \sqsubseteq \mathsf{Person}$ $\mathsf{Man} \sqsubseteq \mathsf{Person}$
ISA05	Import two classes, each subclass of the other $\mathsf{Male} \sqsubseteq \mathsf{Man}$ $\mathsf{Man} \sqsubseteq \mathsf{Male}$
ISA06	Import a class, subclass of itself $\mathsf{Woman} \sqsubseteq \mathsf{Woman}$
ISA07	Import a class which is subclass of an anonymous class defined by an `owl:someValuesFrom` value constraint in an object property $\mathsf{Driver} \sqsubseteq \exists hasCar.\mathsf{Car}$
ISA08	Import a class which is subclass of an anonymous class defined by an `owl:allValuesFrom value` constraint in an object property $\mathsf{Italian} \sqsubseteq \forall wasBorn.\mathsf{Italy}$
ISA09	Import a class which is subclass of an anonymous class defined by an `owl:minCardinality`=0 cardinality constraint in an object property $\mathsf{Employee} \sqsubseteq \geq 0\, worksIn$
ISA10	Import a class which is subclass of an anonymous class defined by an `owl:maxCardinality`=1 cardinality constraint in an object property $\mathsf{Researcher} \sqsubseteq \leq 1\, hasAffiliation$
ISA11	Import a class which is subclass of an anonymous class defined by an `owl:cardinality`=1 cardinality constraint in an object property $\mathsf{Man} \sqsubseteq = 1\, hasMother$
ISA12	Import a class which is subclass of an anonymous class defined by an `owl:minCardinality`=0 and an `owl:maxCardinality`=1 cardinality constraints in an object property $\mathsf{Researcher} \sqsubseteq \geq 0\, hasAffiliation$ $\mathsf{Researcher} \sqsubseteq \leq 1\, hasAffiliation$

(continued on next page)

(continued from previous page)

ISA13	Import a class which is subclass of an anonymous class defined by an `owl:minCardinality`=0 cardinality constraint in a datatype property Person $\sqsubseteq \geq 0\, hasName$
ISA14	Import a class which is subclass of an anonymous class defined by an `owl:maxCardinality`=1 cardinality constraint in a datatype property Researcher $\sqsubseteq \leq 1\, wrotePhDThesis$
ISA15	Import a class which is subclass of an anonymous class defined by an `owl:cardinality`=1 cardinality constraint in a datatype property Person $\sqsubseteq = 1 hasSSN$
ISA16	Import a class which is subclass of an anonymous class defined by an `owl:minCardinality`=0 and an `owl:maxCardinality`=1 cardinality constraints in a datatype property Researcher $\sqsubseteq \geq 0\, wrotePhDThesis$ Researcher $\sqsubseteq \leq 1\, wrotePhDThesis$
ISA17	Import a class which is subclass of a class defined by the intersection of two other classes ItalianMan$\sqsubseteq$ (Italian $\sqcap$ Male)

Group B: Class Equivalences

ISB01	Import several classes which are all of them equivalent Italian $\equiv$ Italiano $\equiv$ Italienne
ISB02	Import a class which is equivalent to an anonymous class defined by an `owl:someValuesFrom` value constraint in an object property Driver$\equiv$ $\exists hasCar.$Car
ISB03	Import a class which is equivalent to an anonymous class defined by an `owl:allValuesFrom` value constraint in an object property Italian$\equiv$ $\forall wasBorn.$Italy
ISB04	Import a class which is equivalent to an anonymous class defined by an `owl:minCardinality`=1 cardinality constraint in an object property Employee$\equiv$ $\geq 1\, worksIn$
ISB05	Import a class which is equivalent to an anonymous class defined by an `owl:maxCardinality`=1 cardinality constraint in an object property Researcher$\equiv$ $\leq 1\, hasAffiliation$
ISB06	Import a class which is equivalent to an anonymous class defined by an `owl:cardinality`=1 cardinality constraint in an object property Man$\equiv$ $= 1\, hasMother$

(continued on next page)

(continued from previous page)

ISB07	Import a class which is equivalent to an anonymous class defined by an `owl:minCardinality`=0 and an `owl:maxCardinality`=1 cardinality constraints in an object property Researcher$\equiv$ $(\leq 1\, hasAffiliation \sqcap \geq 0\, hasAffiliation)$
ISB08	Import a class which is equivalent to an anonymous class defined by an `owl:minCardinality`=0 cardinality constraint in a datatype property Person $\equiv \geq 0\, hasName$
ISB09	Import a class which is equivalent to an anonymous class defined by an `owl:maxCardinality`=1 cardinality constraint in a datatype property Researcher $\equiv \leq 1\, wrotePhDThesis$
ISB10	Import a class which is equivalent to an anonymous class defined by an `owl:cardinality`=1 cardinality constraint in a datatype property Person $\equiv = 1 hasSSN$
ISB11	Import a class which is equivalent to an anonymous class defined by an `owl:minCardinality`=0 and an `owl:maxCardinality`=1 cardinality constraints in a datatype property Researcher $\equiv \geq 0\, wrotePhDThesis$ Researcher $\equiv \leq 1\, wrotePhDThesis$
ISB12	Import a class which is equivalent to an anonymous class defined by the intersection of two other classes ItalianMan$\equiv$(Italian $\sqcap$ Male)

Group C: Class defined by set operators

ISC01	Import a class which is intersection of two other classes ItalianMan $\equiv$(Italian $\sqcap$ Male)
ISC02	Import a class which is intersection of several other classes ItalianMan $\equiv$(Italian $\sqcap$ Male $\sqcap$ Person)

2 Property benchmarks

Group D: Property hierarchies

ISD01	Import a single object property $hasChild$
ISD02	Import an object property that is subproperty of another object property that is subproperty of a third one $isFatherOf \sqsubseteq isGrandFatherOf \sqsubseteq isAncestorOf$

(continued on next page)

(continued from previous page)

ISD03	Import a single datatype property $hasAge$
ISD04	Import a datatype property that is subproperty of another datatype property that is subproperty of a third one $isInteger \sqsubseteq isRational \sqsubseteq isReal$

Group E: Properties with domain and range

ISE01	Import a single object property with domain a class $\top \sqsubseteq \forall hasChild^{-}.\mathsf{Person}$
ISE02	Import a single object property with range a class $\top \sqsubseteq \forall hasChild.\mathsf{Person}$
ISE03	Import a single object property with domain a class and range another class $\top \sqsubseteq \forall hasChild^{-}.\mathsf{Father}$ $\top \sqsubseteq \forall hasChild.\mathsf{Person}$
ISE04	Import a single object property with domain and range the same class $\top \sqsubseteq \forall hasChild^{-}.\mathsf{Person}$ $\top \sqsubseteq \forall hasChild.\mathsf{Person}$
ISE05	Import a single object property with domain multiple classes and range a class $\top \sqsubseteq \forall hasChild^{-}.\mathsf{Mother}$ $\top \sqsubseteq \forall hasChild^{-}.\mathsf{Woman}$ $\top \sqsubseteq \forall hasChild^{-}.\mathsf{Person}$ $\top \sqsubseteq \forall hasChild.\mathsf{Person}$
ISE06	Import a single object property with domain a class and range multiple classes $\top \sqsubseteq \forall hasChild^{-}.\mathsf{Person}$ $\top \sqsubseteq \forall hasChild.\mathsf{Person}$ $\top \sqsubseteq \forall hasChild.\mathsf{Human}$ $\top \sqsubseteq \forall hasChild.\mathsf{Child}$
ISE07	Import a single datatype property with domain a class $\top \sqsubseteq \forall hasSSN^{-}.\mathsf{Person}$
ISE08	Import a single datatype property with range `rdfs:Literal` $\top \sqsubseteq \forall hasName.\texttt{rdfs:Literal}$
ISE09	Import a single datatype property with domain a class and range `rdfs:Literal` $\top \sqsubseteq \forall hasName^{-}.\mathsf{Person}$ $\top \sqsubseteq \forall hasName.\texttt{rdfs:Literal}$

(continued on next page)

(continued from previous page)

ISE10	Import a single datatype property with domain multiple classes and range `rdfs:Literal` $\top \sqsubseteq \forall hasChildNamed^{-}.\mathsf{Mother}$ $\top \sqsubseteq \forall hasChildNamed^{-}.\mathsf{Woman}$ $\top \sqsubseteq \forall hasChildNamed.\texttt{rdfs:Literal}$

Group F: Property equivalences

ISF01	Import several object properties with domain a class and range another class, which are all of them equivalent $\top \sqsubseteq \forall livesIn^{-}.\mathsf{Person}$ $\top \sqsubseteq \forall livesIn.\mathsf{City}$ $livesIn \equiv isResdentIn$
ISF02	Import several datatype properties with domain a class and range `rdfs:Literal`, which are all of them equivalent $\top \sqsubseteq \forall hasName^{-}.\mathsf{City}$ $\top \sqsubseteq \forall hasName.\texttt{rdfs:Literal}$ $hasName \equiv hasSpanishName$
ISF03	Import an object property with domain a class and range another class, which is inverse of another object property $\top \sqsubseteq \forall hasParent^{-}.\mathsf{Child}$ $\top \sqsubseteq \forall hasParent.\mathsf{Person}$ $hasChild \equiv hasParent^{-}$

Group G: Logical characteristics of properties

ISG01	Import a single transitive object property with domain and range the same class $hasFriend^{+} \sqsubseteq hasFriend$ $\top \sqsubseteq \forall hasFriend^{-}.\mathsf{Person}$ $\top \sqsubseteq \forall hasFriend.\mathsf{Person}$
ISG02	Import a single symmetric object property with domain and range the same class $hasFriend \equiv hasFriend^{-}$ $\top \sqsubseteq \forall hasFriend^{-}.\mathsf{Person}$ $\top \sqsubseteq \forall hasFriend.\mathsf{Person}$
ISG03	Import a single functional object property with domain a class and range another class $\top \sqsubseteq \forall hasHusband$ $\top \sqsubseteq \forall hasHusband^{-}.\mathsf{Woman}$ $\top \sqsubseteq \forall hasHusband.\mathsf{Man}$

(continued on next page)

(continued from previous page)

ISG04	Import a single functional datatype property with domain a class and range `rdfs:Literal` $\top \sqsubseteq \forall hasAge$ $\top \sqsubseteq \forall hasAge^{-}.\mathsf{Person}$ $\top \sqsubseteq \forall hasAge.\texttt{rdfs:Literal}$
ISG05	Import a single inverse functional object property with domain a class and range another class $\top \sqsubseteq \forall hasTutor^{-}$ $\top \sqsubseteq \forall hasTutor^{-}.\mathsf{Professor}$ $\top \sqsubseteq \forall hasTutor.\mathsf{Student}$

3 Individual benchmarks

Group H: Single individuals

ISH01	Import one class and one individual that is instance of the class Person(PETER)

ISH02	Import several classes and one individual that is instance of all of them
	Person(PETER)
	Father(PETER)
	Student(PETER)

ISH03	Import one class and several individuals that are instance of the class
	Person(PETER)
	Person(PAUL)
	Person(MARY)

Group I: Named individuals and properties

ISI01	Import one class, one object property with domain and range the class, and one individual of the class that has the object property with another individual of the same class
	$\top \sqsubseteq \forall hasChild^{-}.\mathsf{Person}$
	$\top \sqsubseteq \forall hasChild.\mathsf{Person}$
	Person(MARY)
	Person(PAUL)
	hasChild(MARY, PAUL)

(continued on next page)

(continued from previous page)

ISI02	Import one class, one object property with domain and range the class, and one individual of the class that has the object property with himself $\top \sqsubseteq \forall hasChild^{-}.$Person $\top \sqsubseteq \forall hasChild.$Person Person(PAUL) $knows$(PAUL, PAUL)
ISI03	Import two classes, one object property with domain one class and range the other class, and one individual of one class that has the object property with an individual of the other class $\top \sqsubseteq \forall hasChild^{-}.$Mother $\top \sqsubseteq \forall hasChild.$Child Mother(MARY) Child(PAUL) $hasChild$(MARY, PAUL)
ISI04	Import one class, one datatype property with domain the class and range `rdfs:Literal`, and one individual of the class that has the datatype property with a literal $\top \sqsubseteq \forall hasName^{-}.$Person $\top \sqsubseteq \forall hasName.$`rdfs:Literal` Person(MARYSMITH) hasName(MARYSMITH, “Mary”)
ISI05	Import one class, one datatype property with domain the class and range `rdfs:Literal`, and one individual of the class that has the datatype property with several literals $\top \sqsubseteq \forall hasName^{-}.$Person $\top \sqsubseteq \forall hasName.$`rdfs:Literal` Person(MARYANN) $hasName$(MARYANN, “Mary”) $hasName$(MARYANN, “Ann”)

Group J: Anonymous individuals and properties

ISJ01	Import one class, one object property with domain and range the class, and one anonymous individual of the class that has the object property with another individual of the same class $\top \sqsubseteq \forall hasChild^{-}.$Person $\top \sqsubseteq \forall hasChild.$Person Person(JOHN) $hasChild$(*ANON*[a], JOHN) [a] This denotes an *anonymous indivudual*

(continued on next page)

(continued from previous page)

ISJ02	Import two classes, one object property with domain one class and range the other class, and one anonymous individual of one class that has the datatype property with an individual of the other class $\top \sqsubseteq \forall hasChild^{-}.\mathsf{Parent}$ $\top \sqsubseteq \forall hasChild.\mathsf{Person}$ $\mathsf{Person}(\mathsf{JOHN})$ $hasChild(\mathsf{\mathit{ANON}}, \mathsf{JOHN})$
ISJ03	Import one class, one datatype property with domain the class and range rdfs:Literal, and one anonymous individual of the class that has the datatype property with a literal $\top \sqsubseteq \forall hasName^{-}.\mathsf{Person}$ $\top \sqsubseteq \forall hasName.\mathtt{rdfs{:}Literal}$ $hasName(\mathsf{\mathit{ANON}}, \text{“Peter”})$

Group K: Individual identity

ISK01	Import one class and two named individuals of the class that are the same $\mathsf{Person}(\mathsf{MARYANN}) = \mathsf{Person}(\mathsf{MARY})$
ISK02	Import one class and two named individuals of the class that are different $\neg\Big(\mathsf{Person}(\mathsf{MARYANN}) = \mathsf{Person}(\mathsf{MARY})\Big)$
ISK03	Import one class and three named individuals of the class that are all of them different $\neg\Big(\mathsf{Person}(\mathsf{MARY}) = \mathsf{Person}(\mathsf{ANN})\Big)$ $\neg\Big(\mathsf{Person}(\mathsf{MARY}) = \mathsf{Person}(\mathsf{JOAN})\Big)$ $\neg\Big(\mathsf{Person}(\mathsf{JOAN}) = \mathsf{Person}(\mathsf{ANN})\Big)$

References

[1] Rapahel Volz. *Web ontology reasoning with logic databases.* PhD thesis, AIFB Karlsruhe, 2004.