Rough Subsumption Reasoning with rOWL

C. Maria Keet
School of Computer Science
University of KwaZulu-Natal
Durban, South Africa
keet@ukzn.ac.za

ABSTRACT

There are various recent efforts to broaden applications of ontologies with vague knowledge, motivated in particular by applications of bio(medical)-ontologies, as well as to enhance rough set information systems with a knowledge representation layer by giving more attention to the intension of a rough set. This requires not only representation of vague knowledge but, moreover, reasoning over it to make it interesting for both ontology engineering and rough set information systems. We propose a minor extension to OWL 2 DL, called *rOWL*, and define the novel notions of rough subsumption reasoning and classification for rough concepts and their approximations.

Categories and Subject Descriptors

I.2.4 [**Knowledge Representation Formalisms and Methods**]: Representation languages; F.4 [**Mathematical Logic and Formal Languages**]: Miscellaneous

Keywords

Rough Sets, OWL, Automated Reasoning, Semantic Web

1. INTRODUCTION

Rough set theory and its applications have been shown to be very useful for scenarios where one has to analyse and cope with vague or incomplete data. Compared to fuzzy logics where one already knows the properties and fine-tunes their values, rough set applications enable one to experiment with finding the optimal set of properties of a set of objects in the software system. This approach has the potential to be very useful also in knowledge management, as demonstrated by promising implemented use cases for *in silico* hypothesis testing with bio-ontologies as part of a scientist's research methodology [10] and disease characterisation and patient classification using electronic health record data [19]. To realise rough knowledge management, rough sets have to be integrated with the knowledge representation layer and suitable reasoning services need to be devised with the aim to prove that the rough knowledge is not only consistent but that one also can avail of the taxonomic reasoning to categorise the concepts corresponding to the rough sets. The latter, in turn, can serve as an additional method to find the optimal rough concepts in the rough information system.

To be able to arrive at reasoning services, one first has to devise a sufficiently comprehensive rough ontology language. There are several formalisms of rough sets, such as in DATALOG$^{\neg}$ [4], extended logic programs [21], and Description Logics (DL) with extensions to the Web Ontology Language OWL in particular [2, 5, 7, 8, 12, 13, 19], where each language proposed includes core notions of rough sets only in part [9] and it is not optimised for the most expressive and recently standardised OWL 2 ontology languages that already enjoy substantial implementation infrastructure and user uptake. Reasoning with rough ontologies falls into two categories: reasoning over the instances by querying the data to ascertain if the class is indeed a rough class [10] and type-level reasoning, with the principal reasoning services being possible and definite satisfiability, and rough subsumption reasoning and classification of the rough classes. Thus far, only [8] considers possibly and definitely satisfiable and crisp subsumption of rough concepts for an arbitrary DL language, RDL_{AC}, but neither what can be deduced about the rough concepts from their respective approximations nor enforced on the approximations given an asserted subsumption of rough concepts. To fill this gap and thereby move closer to the realisation of ontology-driven rough knowledge bases, we propose the novel reasoning services of *rough subsumption reasoning* where $C \sqsubseteq D$ cannot be guaranteed in every model, but generally can be expected to hold. To demonstrate this, we extend OWL 2 DL to deal with basic aspects of roughness, called *rOWL* (in such a way that it maintains its well-known properties), and prove the eight permutations for subsumption.

In the remainder of the paper, we provide a recap of the core aspects of rough sets and introduce *rOWL* in Section 2, then we move on to the subsumption reasoning in Section 3, discuss it in Section 4, and conclude in Section 5.

2. ROUGH SETS AND ONTOLOGIES

We briefly summarise the core notions of rough sets and introduce *rOWL*, which approximates theses basic notions.

2.1 Rough sets

The standard Pawlak rough set model consists of an *information system* $I = (U, A)$, where U is a non-empty finite

SAICSIT '11, October 3-5, Cape Town, South Africa

set of objects and A a non-empty finite set of attributes such that for every $a \in A$, we have the function $a : U \mapsto V_a$ where V_a is the set of values that attribute a can have. For any subset of attributes $P \subseteq A$, the equivalence relation $\textsc{ind}(P)$ is defined as

$$\textsc{ind}(P) = \{(x, y) \in U \times U \mid \forall a \in P, a(x) = a(y)\} \quad (1)$$

$\textsc{ind}(P)$ generates a partition of U, which is denoted with $U/\textsc{ind}(P)$, or U/P for short. If $(x, y) \in \textsc{ind}(P)$, then x and y are indistinguishable with respect to the attributes in P, i.e, they are *p-indistinguishable.*

From the objects in universe U, we want to represent set X such that $X \subseteq U$ using the attribute set P where $P \subseteq A$. It may not be possible to represent X in a crisp way—the set may include and/or exclude objects which are indistinguishable on the basis of the attributes in P—but it can be approximated by using lower and upper approximation:

$$\underline{P}X = \{x \mid [x]_P \subseteq X\} \quad (2)$$
$$\overline{P}X = \{x \mid [x]_P \cap X \neq \emptyset\} \quad (3)$$

where $[x]_P$ denotes the equivalence classes of the p-indistinguishability relation. The *lower approximation* (Eq. 2) is the set of objects that are *positively* classified as being members of set X, i.e., it is the union of all equivalence classes in $[x]_P$. The *upper approximation* (Eq. 3) is the set of objects that are *possibly* in X; its complement, $U - \overline{P}X$, is the *negative region* with sets of objects that are definitely not in X (i.e., $\neg X$). Then, "with every rough set we associate two *crisp* sets, called *lower* and *upper approximation*" [18], which is denoted as a tuple $X = \langle \underline{X}, \overline{X} \rangle$. The difference between the lower and upper approximation, $B_P X = \overline{P}X - \underline{P}X$, is the *boundary region* of which its objects neither can be classified as to be members of X nor that they are not in X; if $B_P X = \emptyset$ then X is a crisp set with respect to P and when $B_P X \neq \emptyset$ then X is rough with respect to P. Further, observe that $\underline{P}X \subseteq X \subseteq \overline{P}X$.

These rough set notions are graphically depicted in Figure 1 and illustrated in the following example.

Example 1. Consider Table 1 and attribute set $P = \{\mathsf{Age}, \mathsf{Wheels}, \mathsf{Engine}\}$ as subset of $A = \{\mathsf{Age}, \mathsf{Wheels}, \mathsf{Engine}, \mathsf{Helmet}\}$, then the equivalence classes are:
$[x_{one}] = \{o_1, o_8\}$, $[x_{two}] = \{o_2\}$, $[x_{three}] = \{o_3\}$, $[x_{four}] = \{o_4, o_5\}$, $[x_{five}] = \{o_6, o_7\}$, $[x_{six}] = \{o_9\}$.
Assume our target set is $\{o_3, o_4, o_5, o_6\} \in X$, then $[x_{three}]$ and $[x_{four}]$ are *definitely* in out target set, i.e., their union is the *lower approximation* ($\underline{X} = \{o_3, o_4, o_5\}$). This still misses o_6, for which we need $[x_{five}] = \{o_6, o_7\}$. With P, it is not possible to distinguish between o_6 and o_7, so $[x_{five}]$ as such cannot be part of our target set, i.e., with the given data and selected attributes, there is no way to represent X such that it includes o_6 but excludes o_7. Therefore, the *upper approximation* of X is then the union of the three equivalence classes, being $\overline{X} = \{o_3, o_4, o_5, o_6, o_7\}$, which results in a *boundary region* of $\{o_6, o_7\}$. If P would have contained Helmet, then X were a crisp set. Generally, once there are equivalence classes with more than one object, there likely will be at least one rough set. ♢

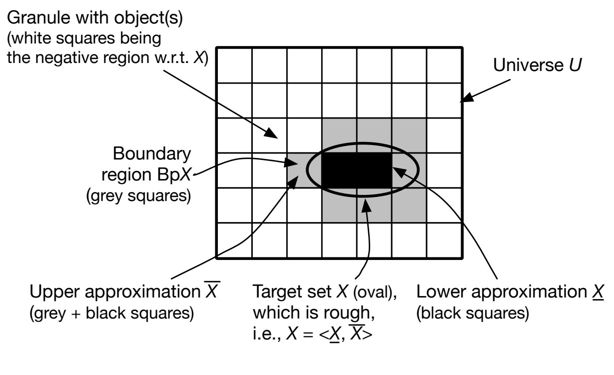

Figure 1: A rough set and associated notions.

Table 1: Sample data table about 9 vehicles.

	Age	Wheels	Engine	Helmet
o_1	< 5	2	no	no
o_2	> 5	2	no	no
o_3	> 5	2	yes	yes
o_4	> 5	3	yes	yes
o_5	> 5	3	yes	yes
o_6	> 5	3	no	yes
o_7	> 5	3	no	no
o_8	< 5	2	no	no
o_9	< 5	4	yes	no

The rough set notions *reduct* and *core* can be considered to be the set of *sufficient* conditions (attributes) and the set of *necessary* conditions, respectively, to maintain the equivalence class structure induced by P. Thus, we have $\textsc{Core} \subseteq \textsc{Red} \subseteq P$ such that $[x]_{\textsc{Red}} = [x]_P$ and $\textsc{Red}$ is minimal for any $a \in \textsc{Red}$ (i.e., $[x]_{\textsc{Red}-\{a\}} \neq [x]_P$), and for any reduct of P, $\textsc{Red}_1, \ldots, \textsc{Red}_n$, the core is its intersection, i.e., $\textsc{Core} = \textsc{Red}_1 \cap \ldots \cap \textsc{Red}_n$.

2.2 A rough ontology language

2.2.1 General considerations

Defining an arbitrary rough ontology language is an interesting exercise, but we shall focus here on the W3C standardised ontology language OWL 2 DL and its underlying Description Logic language $\mathcal{SROIQ}$, for which considerable application infrastructure exists.

Considering related works, earlier formalisations of roughness in DLs omitted either DL roles or OWL's data properties [2, 5, 7, 8, 13, 19], have partial or no relational properties of the indistinguishability relation [2, 5, 12, 13], and/or do not include that the rough concept is a tuple of its lower and upper approximation where all three concepts use the same set of properties [19]. The most inclusive attempt thus far is the arbitrary DL language RDL_{AC} [8] that extends $\mathcal{ALC}$, but its computational complexity has not been proved and it does not propose a solution how to reformulate or approximate the class-as-tuple notation. The identification of a rough class as a tuple can be fully rewritten theoretically, i.e., maintaining logical equivalence, in analogy to a *weak entity type* in the Entity-Relationship conceptual data modelling language [9]. This requires full identification constraints across relationships (DL roles/OWL object properties), which is not possible in OWL 2 DL and requires a costly extension to a language that is already computationally expensive (2NExpTime-complete). However, it can be approximated sufficiently by introducing two new relationships relating the rough class to its approximations explicitly, quantify them existentially, and declare them functional [9]. To this end, we will introduce a `lapr` and `uapr` for each rough class to relate a rough class (domain) to its `l`ower and `u`pper `apr`oximation (range) of the OWL object properties, respectively.

The remaining features mentioned earlier—IND's relational properties, being reflexivity, symmetry, and transitivity—can be represented fully in OWL 2 DL, but the indistinguishability relation and its usage require some clarification. As Eq. 1 indicates, the IND relations is with respect to a *single subset* P of properties and over the *whole* universe. Typically, in rough sets, the 'whole' universe is a large table where each column other than the first denotes an attribute, such as Table 1 about the class Vehicle, but in the setting of an ontology, this may not necessarily be *all* object- and data properties. To clarify this matter in the setting of ontologies and avoid abuse of notation, let:

- A be the set of attributes as in rough sets, i.e., the data properties;
- Π be the set of object- and data properties in the OWL ontology used for declaring the properties of classes;
- P be a subset of A, as in rough sets;
- P_c be a subset of Π for a (rough) class C in the OWL ontology;
- IND be the indistinguishability relation, as in Eq. 1;
- IND_C be the indistinguishability relation as in Eq. 1 regarding class C, where P is replaced with P_c;

They will be defined in the next section.

Thus, aside from the approximation of the identification of the rough class, all features can be represented with OWL 2 DL already, which therewith greatly simplifies the whole endeavour because we already can avail of not only the theoretical results but also its application infrastructures. Although its current technologies are not integrated with scalable large ontologies (TBoxes) or large amounts of data (in the ABox) [9], this is not expected to be problem in the near future thanks to recent efforts in scalable Semantic Web technologies (e.g., [1, 3, 14, 15]). To incorporate the roughness aspects, such as IND and a rough class, we will add a minor extension to fix their syntax and semantics, which will be shown to be useful for subsumption reasoning.

2.2.2 Defining Rough OWL: rOWL

Here we summarise the OWL 2 DL syntax and semantics only insofar as is necessary for rough ontologies; the interested reader is referred to [16] for further details. We extend the vocabulary with notions specific to rough ontologies, being rough class ($\wr C$), lower- and upper approximation ($\underline{C}$ and $\overline{C}$, respectively) that are related to $\wr C$ through the `lapr` and `uapr` object properties to approximate the tuple notation of a rough class, and the reflexive, transitive, and symmetric indistinguishability relation (IND). By standard OWL notation, OP denotes an object property; OPE denotes an object property expression; DP denotes a data property; and C denotes a class.

Definition 1. (Rough OWL 2 DL (rOWL) Ontology Syntax (abbreviated)) A rough vocabulary $V = (V_C, V_{OP}, V_{DP}, V_I, V_{DT}, V_{LT}, V_{FA}, V_\Pi)$ over a datatype map D (as formalised in [16]) is an 8-tuple consisting of the following elements:

- V_C is a set of classes containing at least the classes `owl:Thing` and `owl:Nothing`, and subsets V_{CL} of lower approximations, V_{CU} of upper approximations, and V_{RC} of rough classes;
- V_{OP} is a set of object properties, containing at least
  - the object properties `owl:topObjectProperty` and `owl:bottomObjectProperty`, and
  - `owl:ind`, `owl:lapr`, and `owl:uapr`, and such that we denote with appended subscripts the label of the class these relations apply to (hence, they are sub object properties; e.g., $\texttt{owl:ind}_{car}$ is a subobject property of `owl:ind` where $Car \in V_{RC}$);
- V_{DP} is a set of data properties, containing at least the data properties `owl:topDataProperty` and `owl:bottomDataProperty`;
- V_I is a set of individuals;
- V_{DT} is a set containing all datatypes of D;
- V_{LT} is a set of literals for each datatype and each lexical form;
- V_{FA} is the set of pairs (F, lt) for each constraining facet F, datatype $DT \in N_{DT}$, and literal $lt \in V_{LT}$ such that $(F, (LV, DT_1)LS) \in N_{FS}(DT)$, where LV is the lexical form of lt and DT_1 is the datatype of lt;
- V_Π is the union of the sets of object and data properties, $V^-_{OP} \cup V^-_{DP}$, where V^-_{OP} (resp. V^-_{DP}) are subsets of V_{OP} (resp. V_{DP}) that (*i*) do not contain `owl:ind`, `owl:lapr`, and `owl:uapr` (and its sub-properties), and (*ii*) do not have top and bottom object (data-) properties.

Definition 2. (Rough OWL 2 DL (rOWL) Ontology Semantics (abbreviated)) Given a datatype map D and a vocabulary V over D, an interpretation $I = (\Delta_I, \Delta_D, \cdot^C, \cdot^{OP}, \cdot^{DP}, \cdot^I, \cdot^{DT}, \cdot^{LT}, \cdot^{FA})$ for D and V is a 9-tuple with the following structure:

- Δ_I is a nonempty set called the object domain;
- Δ_D is a nonempty set disjoint with Δ_I called the data domain such that $(DT)^{DT} \subseteq \Delta_D$ for each datatype $DT \in V_{DT}$;
- $\cdot^C$ is the class interpretation function that assigns to each class $C \in V_C$ a subset $(C)^C \subseteq \Delta_I$, and $\cdot^C$ is extended to class expressions as follows
  - `ObjectAllValuesFrom`(OPE CE), $\{x \mid \forall y : (x,y) \in (OPE)^{OP} \text{ implies } y \in (CE)^C\}$;
  - `ObjectSomeValuesFrom`(OPE CE), $\{x \mid \exists y : (x,y) \in (OPE)^{OP} \text{ and } y \in (CE)^C\}$;
  - `ObjectMinCardinality`(n OPE CE), $\{x \mid \sharp\{y \mid (x,y) \in (OPE)^{OP} \text{ and } y \in (CE)^C\} \geq n\}$;
  - `ObjectExactCardinality`(n OPE CE), $\{x \mid \sharp\{y \mid (x,y) \in (OPE)^{OP} \text{ and } y \in (CE)^C\} = n\}$;

  and such that
  - For each $\overline{C} \in V_{CU}$, $(\overline{\mathtt{C}})^C = \{x \in \Delta_I \mid \exists y \in \Delta_I, (x,y) \in (\mathtt{ind}_c)^{OP} \wedge y \in (\mathtt{C})^C\}$ holds,
  - For each $\underline{C} \in V_{CL}$, $(\underline{\mathtt{C}})^C = \{x \in \Delta_I \mid \forall y \in \Delta_I, (x,y) \in (\mathtt{ind}_c)^{OP} \rightarrow y \in (\mathtt{C})^C\}$ holds, and
  - For each $\wr C \in V_{RC}$, $(\wr C)^C = (\langle \underline{C}, \overline{C} \rangle)^C = \langle (\underline{C})^C, (\overline{C})^C \rangle \approx \{x \in \Delta_I \mid \exists x', x'' \in \Delta_I, (x,x') \in (\mathtt{lapr}_c)^{OP} \wedge (x,x'') \in (\mathtt{uapr}_c)^{OP} \wedge x \in (\wr\mathtt{C})^C \wedge x' \in (\underline{\mathtt{C}})^C \wedge x'' \in (\overline{\mathtt{C}})^C\}$ holds;
- $\cdot^{OP}$ is the object property interpretation function that assigns to each object property $OP \in V_{OP}$ a subset $(OP)^{OP} \subseteq \Delta_I \times \Delta_I$ and such that $\cdot^{OP}$ is extended to `ObjectInverseOf`(OP) with the meaning $\{(x,y) \mid (y,x) \in (OP)^{OP}\}$; in addition,
  - $\mathtt{ind} \in V_{OP}$ such that $\mathtt{ind}^{OP} \subseteq \Delta_I \times \Delta_I$ with Re-

flexiveObjectProperty(ind), SymmetricObjectProperty(ind), and TransitiveObjectProperty(ind), and for each rough class, $\wr C \in V_{RC}$, each $\mathtt{ind}_c \in V_{OP}$ such that $\mathtt{ind}_c^{OP} \subseteq (\wr C)^C \times (\wr C)^C$, hence $\mathtt{ind}_c^{OP} \subseteq \mathtt{ind}^{OP}$;

- lapr $\in V_{OP}$ such that $\mathtt{lapr}^{OP} \subseteq (\wr C)^C \times (\underline{\mathtt{C}})^C$;
- uapr $\in V_{OP}$ such that $\mathtt{uapr}^{OP} \subseteq (\wr C)^C \times (\overline{\mathtt{C}})^C$;

- $\cdot^{DP}$ is the data property interpretation function that assigns to each data property $DP \in V_{DP}$ a subset $(DP)^{DP} \subseteq \Delta_I \times \Delta_D$;
- $\cdot^I$ is the individual interpretation function that assigns to each individual $a \in V_I$ an element $(a)^I \in \Delta_I$;
- $\cdot^{DT}$ is the datatype interpretation function that assigns to each datatype $DT \in V_{DT}$ a subset $(DT)^{DT} \subseteq \Delta_D$;
- $\cdot^{LT}$ is the literal interpretation function that is defined as $(lt)^{LT} = (LV, DT)^{LS}$ for each $lt \in V_{LT}$, where LV is the lexical form of lt and DT is the datatype of lt;
- $\cdot^{FA}$ is the facet interpretation function that is defined as $(F, lt)^{FA} = (F, (lt)^{LT})^{FS}$ for each $(F, lt) \in V_{FA}$.

Concerning satisfaction of OWL 2 class expression axioms in interpretation I with respect to ontology O, the class axioms SubClassOf(CE_1 CE_2) holds if $(CE_1)^C \subseteq (CE_1)^C$, EquivalentClasses($\mathrm{CE}_1 \ldots \mathrm{CE}_n$) if $(CE_j)^C = (CE_k)^C$ for each $1 \leq j \leq n$ and each $1 \leq k \leq n$, DisjointClasses($\mathrm{CE}_1 \ldots \mathrm{CE}_n$) if $(CE_j)^C \cap (CE_k)^C = \emptyset$ for each $1 \leq j \leq n$ and each $1 \leq k \leq n$ such that $j \neq k$. Regarding satisfaction of property axioms, SubObjectPropertyOf(OPE_1 OPE_2) holds if $(OPE_1)^{OP} \subseteq (OPE_2)^{OP}$, and for the relevant object property characteristics, we need TransitiveObjectProperty(OPE) for which $\forall x, y, z : (x, y) \in (OPE)^{OP}$ and $(y, z) \in (OPE)^{OP}$ implies $(x, z) \in (OPE)^{OP}$ hold, ReflexiveObjectProperty(OPE) for which $\forall x : x \in \Delta_I$ implies $(x, x) \in (OPE)^{OP}$ holds, and SymmetricObjectProperty(OPE) for which $\forall x, y : (x, y) \in (OPE)^{OP}$ implies $(y, x) \in (OPE)^{OP}$ holds.

OWL 2 DL is based on Description Logics (DL), and therefore we shall use the more concise DL notation in the remainder of the paper. For instance, the statement

SubClassOf(C D)

in OWL 2 DL functional syntax, i.e., all individuals that are a C are also a D, can be represented equivalently in DL as

$\mathtt{C} \sqsubseteq \mathtt{D}$.

ObjectSomeValuesFrom in Definition 2 is the serialised rendering of the existential quantification ($\exists$), an ObjectInverseOf(OP) is denoted as $\mathtt{OP}^-$, EquivalentClasses(CE_1 CE_2) as $\mathtt{CE}_1 \doteq \mathtt{CE}_2$, and ClassAssertion as C(a). Thus, we have for, e.g., each $\overline{\mathtt{C}}$ and $\underline{\mathtt{C}}$ in the rough ontology, in OWL 2 DL functional syntax:

EquivalentClasses($\overline{\mathtt{C}}$ ObjectSomeValuesFrom(a:ind a:C)),

EquivalentClasses($\underline{\mathtt{C}}$ ObjectAllValuesFrom(a:ind a:C)),

i.e., $\overline{\mathtt{C}} \equiv \exists\mathtt{ind}.\mathtt{C}$ and $\underline{\mathtt{C}} \equiv \forall\mathtt{ind}.\mathtt{C}$ in DL notation.

2.2.3 Notes on *rOWL*

There are three points with respect to *rOWL* worth elaborating on as clarification on its usage.

First, recollect that $\wr C$ is approximated by using uapr and lapr that have $\wr$C as domain and cardinality exactly 1; the assertions for each combination of rough class and its upper and lower approximation is then:

ObjectPropertyDomain(a:upar a:$\wr$C),

ObjectPropertyDomain(a:lapr a:$\wr$C),

ObjectExactCardinality(1 a:uapr a:$\overline{\mathtt{C}}$), and

ObjectExactCardinality(1 a:lapr a:$\underline{\mathtt{C}}$),

i.e., $\exists\mathtt{uapr}^- \sqsubseteq \wr\mathtt{C}$, $\exists\mathtt{lapr}^- \sqsubseteq \wr\mathtt{C}$, $\wr\mathtt{C} \sqsubseteq = 1\ \mathtt{uapr}.\overline{\mathtt{C}}$, and $\wr\mathtt{C} \sqsubseteq = 1\ \mathtt{lapr}.\overline{\mathtt{C}}$, which are added to the ontology for *each* rough concept, its approximations, and indistinguishability relation (justification for this encoding is discussed in [9]).

Second, the semantics for V_Π follows from its definition, given that $V_\Pi \subset V_{OP} \cup V_{DP}$. Likewise, with P_i (where i denotes the label/name of the rough class) such that $P_i \subseteq V_\Pi$, the semantics follows trivially from it. Note that the properties in P_i to compute the equivalence structure are existentially quantified for each object in $\wr C$ (which does not preclude $\wr C$ from having universally quantified properties). Those properties, if any, that are in P_i but not in any of the reducts—thus being superfluous for the equivalence structure—ideally should not be included as properties of $\wr C$ in the ontology, because they do not contribute to the identifying characteristics of the set, hence, not to the class corresponding to the rough set. Nevertheless, they may be useful to include for examining the data as it may turn out that they are mandatory for a $C' \sqsubseteq C$ or a sister-class of C that shares a direct common subsumer with C, depending on the available data or represented knowledge.

A third observation regarding properties and attributes, is that in Pawlak's original information system $I = (U, A)$ only attributes are used, in the sense of OWL's data properties. However, in an OWL ontology, the intension of such 'attributes' from the data tables can be represented with either object properties or data properties, hence the $V_\Pi = V_{OP}^- \cup V_{DP}^-$ instead of a possible more restricted mapping of $A = V_{DP}^-$. To see the usefulness of this, let us take the 'attribute' Wheels from Example 1, which can be represented in an OWL ontology with a DP, say, hasWheels with as domain Bicycle and a data type integer set to 2, e.g.,

$\mathtt{Bicycle} \sqsubseteq \exists\mathtt{hasWheels}.\mathtt{integer}_{=2}$

or one can represent it with an OP hasPart that has Wheel as range and the number as a cardinality constraint, e.g.,

$\mathtt{Bicycle} \sqsubseteq =2\ \mathtt{hasPart}.\mathtt{Wheel}$.

The debate on the merits of one representation over the other in ontology development is long and inconclusive and here is not the place to argue in either direction, just to note that both approaches can, and are being, used in domain ontologies and that one can use either one with *rOWL*.

3. ROUGH SUBSUMPTION REASONING

The so-called 'standard' inferencing problems and reasoning services offered by the DL and OWL reasoners for all scenarios of ontology usage are ontology consistency, class expression satisfiability, class expression subsumption, and instance checking [16]. The idea of rough knowledge base satisfiability (without rough classes) has been introduced earlier [13]. Jiang et al. [8] recently have proposed a *definitely satisfiable concept*—iff $(\underline{C})^\mathcal{I}$ is satisfiable—and a *possibly satisfiable concept*—iff $(\overline{C})^\mathcal{I}$ is satisfiable. More interesting, they have shown the idea of *crisp subsumption of rough concepts* proving it using $\mathcal{O} \models \wr C \sqsubseteq \wr D$ iff $(\overline{C} \sqcap \neg\overline{D})^\mathcal{I}$ is unsatisfiable in their arbitrary RDL_{AC} DL language. A natural next step is to investigate *rough subsumption of crisp/rough concepts*. To arrive at that point, let us first recollect class subsumption in OWL 2:

Definition 3. (Class Expression Subsumption [16]) CE_1 is *subsumed* by a class expression CE_2 w.r.t. O and D if $(\mathrm{CE}_1)^C \subseteq (\mathrm{CE}_2)^C$ for each model $I = (\Delta_I, \Delta_D, \cdot^C, \cdot^{OP}, \cdot^{DP}, \cdot^I, \cdot^{DT}, \cdot^{LT}, \cdot^{FA})$ of O w.r.t. D and V.

This means that for $\texttt{CE}_1 \sqsubseteq \texttt{CE}_2$ to hold in the ontology, the properties of class C_1 in the class expression $\texttt{CE}_1$ must be either more in number than those of C_2 in the class expression $\texttt{CE}_2$ or at least one range of those properties must be more restrictive, or is universally quantified and that property for C_2 is existentially quantified. The latter option is demonstrated already with the rough class and the definition of its approximations: with respect to reasoning over OWL or DL ontologies, $\underline{C} \sqsubseteq \wr C \sqsubseteq \overline{C}$ will be deduced already with a standard non-rough automated reasoner thanks to the existential and universal restriction on IND for $\overline{C}$ and $\underline{C}$, respectively. So, the interesting aspect is going to be the treatment of the properties, in particular because that is also the main strength of application scenarios for rough ontologies.

More precisely, we introduce the novel reasoning services of *definite* and *rough subsumption* by, given the properties, taking into account the interactions between the lower and upper approximations of two rough concepts. That is, we are interested in answering the question *is the extension of C contained in the extension of D in every model of ontology O?* In order to answer this, we have to address two principally different cases:

A. If $\wr C \sqsubseteq \wr D$ is *asserted* in $\mathcal{O}$, what can be said about the subsumption relations among their respective approximations?

B. Given a subsumption between any of the lower and upper approximations of C and D, then can one deduce $\wr C \sqsubseteq \wr D$?

Because being rough or not depends entirely on the chosen properties P_c for class C together with the available data, should these two cases be solved only at the TBox level or necessarily include the ABox for it to make sense? And, in the same line of thinking, should that be under the assumption of standard instantiation and instance checking, or in the presence of a novel DL notion of *rough instantiation* and *rough instance checking*? The latter is not particularly interesting, because the extension is straight-forward to define and has no interesting consequences: with respect to satisfaction in OWL [16], the axiom $\texttt{ClassAssertion}$(CE a) means $(a)^I \in (CE)^C$ so that, by the definition of rough class and its approximations, a rough instantiation or rough class assertion (cf. crisp) is one where $(a)^I \in (\overline{C})^C$ and $(\overline{C})^C - (\underline{C})^C \neq \emptyset$ holds (cf. the crisp $(a)^I \in (\underline{C})^C$). (It makes sense to add the "$(\overline{C})^C - (\underline{C})^C \neq \emptyset$" constraint, because it ensures C is a rough class; if $(\overline{C})^C - (\underline{C})^C = \emptyset$, then the boundary region is empty, hence, C is a crisp set with normal instantiation.) Consequently, one can amend OWL's standard instance checking to:

Definition 4. (Rough Instance Checking) a is an instance of CE w.r.t. O and D if $(a)^I \in (\overline{C})^C$ and $(\overline{C})^C - (\underline{C})^C \neq \emptyset$ for each model $I = (\Delta_I, \Delta_D, \cdot^C, \cdot^{OP}, \cdot^{DP}, \cdot^I, \cdot^{DT}, \cdot^{LT}, \cdot^{FA})$ of O w.r.t. D and V.

Concerning items A and B, similar questions have been asked before for rough sets, but they were limited to what $X \subseteq Y$ implies—and even then the answers differ [18, 11, 8, 22]. [11, 8, 22] assert that

$$X \subseteq Y \Rightarrow \underline{X} \subseteq \underline{Y} \text{ and } \overline{X} \subseteq \overline{Y} \tag{4}$$

for *any* subsets $X, Y \subseteq U$, whereas [18] assert this *only if* the set of attributes $P \subseteq A$ for X and Y are the same:

$$X \subseteq Y \Rightarrow \underline{P}X \subseteq \underline{P}Y \text{ and } \overline{P}X \subseteq \overline{P}Y \tag{5}$$

As we shall see, the latter is correct. Given that subsumption reasoning is considered a crucial reasoning service in both OWL and DL in general, we shall go through all permutations and prove them. Observe hereby that, because we have introduced $\underline{C}$ and $\overline{C}$ already explicitly in the ontology and defined them to be an equivalent class to $\forall\texttt{ind.C}$ and $\exists\texttt{ind.C}$, respectively, we do not need Jiang et al's explicit translation function [8]. In addition, because we have flattened out the rough-class-as-binary-tuple to a class with two properties relating to its respective approximations, we do not need to consider tuple subsumption here, but can avail of the more straight-forward notion of class subsumption.

3.1 Subsumption and properties

Let us first demonstrate the case that Eq. 4 does *not* hold.

PROPOSITION 1. *Let $\mathcal{O}$ be a rOWL ontology, $\wr C$ and $\wr D$ two rough concepts such that $\wr C \sqsubseteq \wr D$ is asserted in the ontology, and $\underline{C}, \underline{D}, \overline{C}, \overline{D}$ are their respective approximations based on* arbitrary *sets of declared properties P_c and P_d (with $P_c, P_d \subseteq V_{\Pi}$), then*
i. $\underline{C} \sqsubseteq \overline{D}$ holds;
ii. $\overline{C} \sqsubseteq \underline{D}$ does not necessarily hold;
iii. $\overline{C} \sqsubseteq \overline{D}$ does not necessarily hold;
iv. $\underline{C} \sqsubseteq \underline{D}$ does not necessarily hold;

PROOF. The proof for (*i*) follows the same line as Jiang et al's subsumption of rough concepts (Theorem 5 in [8]).

To prove (*ii-iv*), it suffices to prove the weakest case cannot be deduced, i.e., (*iv*), which we do by providing a counter example where $x \in (\underline{C})^{\mathcal{I}}$ and $x \notin (\underline{D})^{\mathcal{I}}$. Consider the following ABox:

$$\overline{C}(o_1), \overline{C}(o_2), \overline{C}(o_3), \overline{C}(o_4), \overline{C}(o_5) \tag{6}$$

$$C(o_1), C(o_2), C(o_3), C(o_4) \tag{7}$$

$$\underline{C}(o_1), \underline{C}(o_2), \underline{C}(o_3) \tag{8}$$

$$\underline{D}(o_1), \underline{D}(o_2) \tag{9}$$

$$D(o_1), D(o_2), D(o_3), D(o_4), D(o_5) \tag{10}$$

$$\overline{D}(o_1), \overline{D}(o_2), \overline{D}(o_3), \overline{D}(o_4), \overline{D}(o_5), \overline{D}(o_6) \tag{11}$$

The extension of $\underline{D}$ is a subset of the extension of $\underline{C}$, hence contradicting that $\underline{C} \sqsubseteq \underline{D}$ holds in every model. For (*iii*), it is equally trivial to describe a counter example: add, $\overline{C}(o_6)$ and $\overline{C}(o_7)$, which violates $\overline{C} \sqsubseteq \overline{D}$ due to $\overline{C}(o_7)$. □

The situation above can arise due to the fact that C and D may have different, and, moreover, *incompatible*, properties or incompatible ranges of those properties; that is, $P_c \cap P_d = \emptyset$ and $P_d \subset P_c$ are legal states in a *rough set* setting. Indeed, at the *knowledge representation* layer, adding $\wr C \sqsubseteq \wr D$ with their attributes such that $P_c \cap P_d = \emptyset$, does not hold and is not even possible in ontology development tools such as Protégé, because the ontology development environments force $\wr C$ to inherit the properties declared for $\wr D$, i.e., by asserting $\wr C \sqsubseteq \wr D$ in the ontology, one must have $P_c \cap P_d \neq \emptyset$, provided $P_c \neq \emptyset$ and $P_d \neq \emptyset$, and that $P_d \subseteq P_c$. The 'tricky' aspect is that even when $P_d \subseteq P_c$ holds, they may have incompatible ranges so that they *generate different equivalent structures*, hence, that the basic tenet of Proposition 1 holds. If the *explicitly declared* properties of $\wr C$ or $\wr D$

are such that $P_c \cap P_d = \emptyset$, then $\wr C$ or $\wr D$ will be reclassified to somewhere else in the taxonomy, using a common subsumption reasoning algorithm as implemented in reasoners such as Hermit, Racer, Pellet or Fact++ [6, 17].

One can *add* $\overline{C} \sqsubseteq \overline{D}$ to $\mathcal{O}$ as a constraint on the admissible models. If one were to do so with (6)-(11) together with $\overline{C}(o_6)$ and $\overline{C}(o_7)$ as instantiation of an ABox, then an OWL reasoner will infer $\overline{D}(o_7)$ in order to keep the knowledge base consistent. If it were to be explicitly added that o_7 is *not* an instance of D by adding $D(\neg o_7)$ (for case (*iv*), the assertion $D(\neg o_3)$), then the ontology is, indeed, inconsistent.

Under the same set of attributes and their values/ranges, i.e., $P_c = P_d$, then $\overline{C} \sqsubseteq \overline{D}$ and $\underline{C} \sqsubseteq \underline{D}$ do hold, because in that case a model like (6)-(11) cannot be constructed.

PROPOSITION 2. *Let $\mathcal{O}$ be a rOWL ontology, $\wr C$ and $\wr D$ two rough concepts such that $\mathcal{O} \models \wr C \sqsubseteq \wr D$, $\wr C$ and $\wr D$ have been computed using the* same set of object- and data properties *($P_c = P_d$, with $P_c, P_d \subseteq V_\Pi$) with the* same range restrictions, *and $\underline{C}, \underline{D}, \overline{C}, \overline{D}$ are their respective approximations, then*
i. $\underline{C} \sqsubseteq \overline{D}$ holds;
ii. $\overline{C} \sqsubseteq \underline{D}$ does not necessarily hold;
iii. $\overline{C} \sqsubseteq \overline{D}$ holds;
iv. $\underline{C} \sqsubseteq \underline{D}$ holds;

PROOF. The proof for (*i*) can be constructed in similar fashion to Theorem 5 in [8].

For case (*iv*), consider again the instances (6-11), the semantics for $\underline{C}$ and $\overline{C}$, and some attributes for both $\wr C$ and $\wr D$, such as $\wr \mathtt{D} \sqsubseteq \exists \mathtt{R.E}$ where $\mathtt{E} \in V_C$—hence, also $\wr \mathtt{C} \sqsubseteq \exists \mathtt{R.E}$ holds thanks to $\wr \mathtt{C} \sqsubseteq \wr \mathtt{D}$—and corresponding assertions for the individuals ($\mathtt{R}(\mathtt{o}_1, \mathtt{e}_1)$ etc.). With the same attribute set for $\wr C$ and $\wr D$, i.e., $P_c = P_d$, where $P_c, P_d \subseteq V_\Pi$, it is *guaranteed that the same equivalence structure is generated.* With $\wr C \sqsubseteq \wr D$, then the union of equivalence classes that make up $\underline{C}$ can only consist of a union that is equal or less than that of $\underline{D}$, hence a situation like (8) cf. (9) cannot occur. The reverse holds for $\overline{C} \sqsubseteq \overline{D}$ (case (*iii*)).

For case (*ii*), one can construct a counter example even under the condition of using the same set of attributes, i.e., where $x \in (\overline{C})^\mathcal{I}$ and $x \notin (\underline{D})^\mathcal{I}$ and considering the equivalence classes. Let $\wr C \sqsubseteq \wr D$, and therefore $\overline{C} \sqsubseteq \overline{D}$ and $\underline{C} \sqsubseteq \underline{D}$ also hold. Consider the following model/ABox:

$$\overline{C}(o_1), \overline{C}(o_2), \overline{C}(o_3), \overline{C}(o_4), \overline{C}(o_5) \tag{12}$$
$$C(o_1), C(o_2), C(o_3), C(o_4) \tag{13}$$
$$\underline{C}(o_1), \underline{C}(o_2) \tag{14}$$
$$\underline{D}(o_1), \underline{D}(o_2), \underline{D}(o_3) \tag{15}$$
$$D(o_1), D(o_2), D(o_3), D(o_4), D(o_5), D(o_6) \tag{16}$$
$$\overline{D}(o_1), \overline{D}(o_2), \overline{D}(o_3), \overline{D}(o_4), \overline{D}(o_5), \overline{D}(o_6), \overline{D}(o_7) \tag{17}$$

Take some properties $P_{c,d} \in V_\Pi$, which generate the following equivalence structure induced by P: $[x_{one}] = \{o_1, o_2\}$, $[x_{two}] = \{o_3\}$, $[x_{three}] = \{o_4, o_5\}$, $[x_{four}] = \{o_6, o_7\}$ (a data table can be constructed trivially to match the equivalence structure, and therefore omitted here), which holds for both C and D and their respective roughness. Thus, $\overline{C}$ consists of the union of $[x_{one}]$, $[x_{two}]$, and $[x_{three}]$, whereas $\underline{D}$ consists of the union of just $[x_{one}]$ and $[x_{two}]$. Hence, $\overline{C}$'s extension has more members than $\underline{D}$. □

Having clarified the matter for implication given $\wr C \sqsubseteq \wr D$, we now proceed to rough subsumption.

3.2 Rough Subsumption

Regarding the second case—whether one can derive $\wr C \sqsubseteq \wr D$ given a subsumption relation between any of the approximations—then only if $\overline{C} \sqsubseteq \underline{D}$ then one can deduce $\wr C \sqsubseteq \wr D$, but this cannot be guaranteed for the other three combinations, but we can say that $\wr D$ *roughly subsumes* $\wr C$, denoted with $\wr C \lesssim \wr D$.

PROPOSITION 3. *Let $\mathcal{O}$ be a rOWL ontology, $\underline{C}, \underline{D}, \overline{C}, \overline{D}$ be the lower and upper approximations of $\wr C$ and $\wr D$, then:*
i. if $\underline{C} \sqsubseteq \overline{D}$ then one cannot deduce $\wr C \sqsubseteq \wr D$, but $\wr D$ may roughly subsume *$\wr C$, i.e., $\wr C \lesssim \wr D$;*
ii. if $\overline{C} \sqsubseteq \underline{D}$ then one can deduce $\wr C \sqsubseteq \wr D$;
iii. if $\overline{C} \sqsubseteq \overline{D}$ then one cannot deduce $\wr C \sqsubseteq \wr D$, but $\wr D$ may roughly subsume *$\wr C$, i.e., $\wr C \lesssim \wr D$;*
iv. if $\underline{C} \sqsubseteq \underline{D}$ then one cannot deduce $\wr C \sqsubseteq \wr D$, but $\wr D$ may roughly subsume *$\wr C$, i.e., $\wr C \lesssim \wr D$;*

PROOF. (*ii*) is trivial: all objects possibly in C are definitely in D, hence the extension of $\wr C$—by definition equal or less than $\overline{C}$—is always a subset of D, hence $\wr C \sqsubseteq \wr D$.

Consider (*i*) with two concepts such that $\underline{C} \sqsubseteq \overline{D}$ holds with respect to their instances:

$$\overline{C}(o_1), \overline{C}(o_2), \overline{C}(o_3), \overline{C}(o_4), \overline{C}(o_5) \tag{18}$$
$$C(o_1), C(o_2), C(o_3), C(o_4) \tag{19}$$
$$\underline{C}(o_1), \underline{C}(o_2) \tag{20}$$
$$\underline{D}(o_1), \underline{D}(o_2) \tag{21}$$
$$D(o_1), D(o_2), D(o_3) \tag{22}$$
$$\overline{D}(o_1), \overline{D}(o_2), \overline{D}(o_3) \tag{23}$$

$\wr C \sqsubseteq \wr D$ clearly does not hold, due to $C(o_4)$. Based on (18)-(23) and including $D(\neg o_4)$, then any DL reasoner should deduce $\wr D \sqsubseteq \wr C$, hence reorder the hierarchy in the TBox. If the two rough concepts would have been computed with the same set of attributes, one still cannot guarantee $\wr C \sqsubseteq \wr D$ holds in every model: take some $P_{c,d}$, then an equivalence structure generated on the objects in (18)-(23) can result in $[x_{one}] = \{o_1, o_2\}$, $[x_{two}] = \{o_3\}$, and $[x_{three}] = \{o_4, o_5\}$ for both concepts, yet C is not subsumed by D. However, because this situation occurs *only if* D and $\overline{D}$ have the same set of instances, we can say that $\wr C \lesssim \wr D$.

Consider (*iv*) with two arbitrary concepts such that $\underline{C} \sqsubseteq \underline{D}$ holds with respect to their instances, and the following admissible model or actual instances declared in the ABox:

$$\overline{C}(o_1), \overline{C}(o_2), \overline{C}(o_3), \overline{C}(o_4), \overline{C}(o_5), \overline{C}(o_6) \tag{24}$$
$$C(o_1), C(o_2), C(o_3), C(o_4), C(o_5) \tag{25}$$
$$\underline{C}(o_1), \underline{C}(o_2) \tag{26}$$
$$\underline{D}(o_1), \underline{D}(o_2), \underline{D}(o_3) \tag{27}$$
$$D(o_1), D(o_2), D(o_3), D(o_4) \tag{28}$$
$$\overline{D}(o_1), \overline{D}(o_2), \overline{D}(o_3), \overline{D}(o_4) \tag{29}$$

Again, $\wr C \sqsubseteq \wr D$ does not hold with respect to the model. If $P_c = P_d$, i.e., the equivalence structure generated on the objects is the same for both concepts, then $\wr C \sqsubseteq \wr D$ still does not hold; e.g., $[x_{one}] = \{o_1, o_2\}$, $[x_{two}] = \{o_3\}$, $[x_{three}] = \{o_4\}$, and $[x_{five}] = \{o_5, o_6\}$. However, because this situation occurs *only if* D and $\overline{D}$ have the same set of instances, we can say that $\wr C \lesssim \wr D$.

For case (*iii*) with two arbitrary concepts, the argument

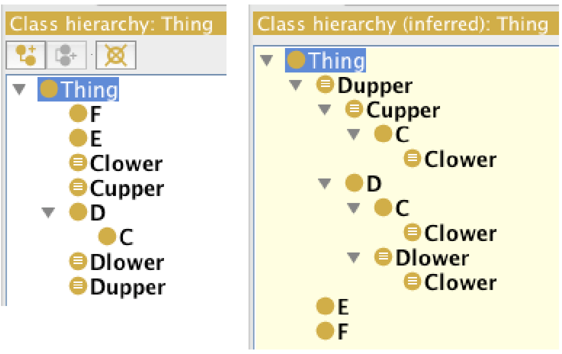

Figure 2: Left: the sloppy taxonomy with all required axioms (see text for details); Right: the inferred hierarchy, which shows that (*i*), (*iii*), and (*iv*) of Proposition 2 hold, but not (*ii*).

is similar to (*iv*), e.g., with:

$$\overline{C}(o_1), \overline{C}(o_2), \overline{C}(o_3), \overline{C}(o_4), \overline{C}(o_5) \quad (30)$$

$$C(o_1), C(o_2), C(o_3), C(o_4), C(o_5) \quad (31)$$

$$\underline{C}(o_1), \underline{C}(o_2) \quad (32)$$

$$\underline{D}(o_1), \underline{D}(o_2), \underline{D}(o_3) \quad (33)$$

$$D(o_1), D(o_2), D(o_3), D(o_4) \quad (34)$$

$$\overline{D}(o_1), \overline{D}(o_2), \overline{D}(o_3), \overline{D}(o_4), \overline{D}(o_5), \overline{D}(o_6) \quad (35)$$

If $P_c = P_d$ then still $\wr C \sqsubseteq \wr D$ does not hold: consider again (30-35) and an equivalence structure $[x_{one}] = \{o_1, o_2\}$, $[x_{two}] = \{o_3\}$, and $[x_{three}] = \{o_4, o_5, o_6\}$, then, because C has o_4, o_5 as members, $\overline{C}$ therefore is forced to have o_6 as member as well due to $[x_{three}]$[1], but while adding o_6 to $\overline{C}$ still satisfies $\overline{C} \sqsubseteq \overline{D}$, it still does not hold that $\wr C \sqsubseteq \wr D$ in this model, hence not in every model. However, because this situation occurs *only if* D has fewr instances declared explicitly than C (and $\overline{D}$ more than $\overline{C}$), we can say that $\wr C \lesssim \wr D$. □

4. DISCUSSION

Although it cannot be guaranteed that $\wr C \sqsubseteq \wr D$ holds for cases (*i*), (*iii*), and (*iv*) in Proposition 3 based on the declared knowledge in the ontology, it will be an interesting avenue to devise an algorithm that checks solely if it holds against the actual ABox and orders the respective classes in the taxonomy accordingly. However, this is beyond the current scope and scalability of Semantic Web technologies.

Observe, though, that while the notion of rough subsumption is not included in any of the OWL reasoners, the others in Propositions 2 and 3 can be obtained already with the standard reasoners (Proposition 1 is not relevant in an OWL setting and therefore not discussed here). To illustrate this, let us take Proposition 2 with $\wr C \sqsubseteq \wr D$, subsumption between their respective approximations using `ind` to relate the approximations to the rough class, `lapr` and `uapr` to relate the rough class to its respective approximations, two arbitrary properties with E and F as range, respectively, and an ABox as in (12-17), encode it in an OWL ontology and run the (crisp) classification reasoning service. This was carried out with Protégé 4.1beta and its built-in reasoner HermiT v1.2.4. The result is shown in Figure 2, demonstrating that (*ii*) of Propositions 2 indeed does not hold, whereas (*i*), (*iii*), and (*iv*) do. This sample OWL ontology and a sample ontology for Proposition 3 to demonstrate the behaviour of standard OWL infrastructure with the crisp reasoning services are available in the online supplementary material at `http://www.meteck.org/files/roughontosuppl/roughontotests.html`.

Caution has to be taken if one wants to use rough OWL ontologies with standard reasoners in the presence of instances, because OWL does not adhere to the unique name assumption. The consequence is that individuals have to be declared disjoint explicitly in order to prevent the reasoner deducing that individuals with different names but having the same values for the declared properties are the same individual. Without the assertion that the individuals are disjoint and where more than one object is a member of an equivalence class, then, depending on the other knowledge represented in the ontology, it may suggest incorrectly that there are only crisp classes (for then it may deduce that each equivalence class has only one member, and thus the boundary region would always be empty).

What may be an interesting avenue to explore is to *enforce* in the ontology development software that either $P_d \subset P_c$ or that the ranges of at least one of the properties in P_c must be a subset of those in P_d. This is illustrated in Example 2.

Example 2. Recollect the vehicles from Example 1 and Table 1, there may be a TBox statement
$\wr$`Vehicle` $\sqsubseteq \exists$`HASNRWHEEL.Integer`
but one may want to define other classes in the ontology, such as
`Tricycle` $\sqsubseteq \exists$`HASNRWHEEL.Integer`$_{=3}$,
hence,
`Tricycle` $\sqsubseteq \wr$`Vehicle`
an examine its roughness. This reduces the equivalence structure to $[x_{four}] = \{o_4, o_5\}$, $[x_{five}] = \{o_6, o_7\}$, which would have been a crisp class if `Tricycle` were a defined class. Observe that this only works if $P_d \subset P_c$. If $P_c = P_d$ and the properties are necessary *and* sufficient, i.e., the properties are in the reduct and core (e.g., `D` $\equiv \exists$`R.E` $\sqcap \exists$`S.F` instead of `D` $\sqsubseteq \exists$`R.E` $\sqcap \exists$`S.F`), the hierarchy collapses to equivalences such that `C` $\equiv$ `D`, $\underline{\texttt{C}} \equiv \underline{\texttt{D}}$, and $\overline{\texttt{C}} \equiv \overline{\texttt{D}}$.
Adding `MotorisedTricycle` to the TBox as
`MotorisedTricycle` $\sqsubseteq$ `Tricycle` $\sqcap$
$\exists$`HASNRWHEEL.Integer`$_{=3}$ $\sqcap$ $\exists$`HASENGINE.Boolean`$_{yes}$
reduces it to a single equivalence class of $[x_{four}] = \{o_4, o_5\}$, that cannot be disambiguated any further, even if we were to include helmet in P_c. But, again, making it a defined class forces it to be a crisp class—or: rough with an empty boundary—with respect to the data. ◇

In this way, the vagueness can be reduced or even 'pushed' out of the ontology thanks to an iterative analysis of the represented knowledge in conjunction with hypothesis testing to find the most suitable properties of the target classes. We are looking into a methodological approach to achieve this in conjunction with the reduct and core.

5. CONCLUSIONS

A rough extension to OWL 2 DL, called *rOWL*, was introduced in such a way that OWL 2 DL's computational

[1]Observe that because $\overline{C}$ is computed from P_c, one cannot add the assertion $\overline{C}(\neg o_6)$ like one can do for C.

complexity was maintained. The novel notions of rough subsumption reasoning and classification for rough concepts and their approximations were defined using *rOWL*. The roughness aspect lies in the fact that the subsumption cannot be guaranteed in every model, but generally can be expected to hold as the violating exception is a peculiar state where one of the target classes and its upper approximation have the same set of instances.

From an engineering and usability perspective, we are currently looking into the usage of properties with rough sets and rough ontologies, and in particular how the reduct and core can be integrated in (rough) ontology engineering and the hypothesis testing scenario of bio-ontologists. From logics point of view, there are multiple avenues one can pursue, as little research has been carried out on rough logics. There are preliminary results with paraconsistent rough DLs [20] and we are looking into its relation with non-monotonic logics and reasoning.

6. REFERENCES

[1] F. Baader, M. Bienvenu, C. Lutz, and F. Wolter. Query and predicate emptiness in description logics. In *12th International Conference on Principles of Knowledge Representation and Reasoning (KR'10)*. AAAI Press, 2010. 9-11 May 2010, Toronto, Canada.

[2] F. Bobillo and U. Straccia. Supporting fuzzy rough sets in fuzzy description logics. In C. Sossai and G. Chemello, editors, *Proceedings of the European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'09)*, volume 5590 of *LNCS*, pages 676–687. Springer, 2009.

[3] D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, A. Poggi, M. Rodríguez-Muro, and R. Rosati. Ontologies and databases: The DL-Lite approach. In S. Tessaris and E. Franconi, editors, *Semantic Technologies for Informations Systems - 5th Int. Reasoning Web Summer School (RW 2009)*, volume 5689 of *LNCS*, pages 255–356. Springer, 2009.

[4] P. Doherty, W. Lukaszewicz, A. Skowron, and A. Szalas. *Knowledge Representation Techniques: A Rough Set Approach*. Studies in fuzziness and soft computing. Springer, 2006.

[5] N. Fanizzi, C. D'Amato, F. Esposito, and T. Lukasiewicz. Representing uncertain concepts in rough description logics via contextual indiscernibility relations. In *Proceedings of the 4th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW'08)*, volume 423 of *CEUR-WS*, 2008.

[6] V. Haarslev and R. Möller. Racer system description. In R. Goré, A. Leitsch, and T. Nipkow, editors, *International Joint Conference on Automated Reasoning, IJCAR'2001, June 18-23, Siena, Italy*, pages 701–705. Springer-Verlag, 2001.

[7] S. Ishizu, A. Gehrmann, Y. Nagai, and Y. Inukai. Rough ontology: extension of ontologies by rough sets. In M. J. Smith and G. Salvendy, editors, *Proceedings of Human Interface and the Management of Information. Methods, Techniques and Tools in Informaiton Design*, volume 4557 of *LNCS*, pages 456–462. Springer, 2007.

[8] Y. Jiang, J. Wang, S. Tang, and B. Xiao. Reasoning with rough description logics: An approximate concepts approach. *Information Sciences*, 179:600–612, 2009.

[9] C. M. Keet. On the feasibility of description logic knowledge bases with rough concepts and vague instances. In *Proceedings of the 23rd International Workshop on Description Logics (DL'10)*, CEUR-WS, pages 314–324, 2010. 4-7 May 2010, Waterloo, Canada.

[10] C. M. Keet. Ontology engineering with rough concepts and instances. In P. Cimiano and H. Pinto, editors, *17th International Conference on Knowledge Engineering and Knowledge Management (EKAW'10)*, volume 6317 of *LNCS*, pages 507–517. Springer, 2010. 11-15 October 2010, Lisbon, Portugal.

[11] M. A. Khan and M. Banerjee. Formal reasoning with rough sets in multiple-source approximation systems. *International Journal of Approximate Reasoning*, 49(2):466 – 477, 2008. Special Section on Probabilistic Rough Sets and Special Section on PGM'06.

[12] M. C. Klein, P. Mika, and S. Schlobach. Rough description logics for modeling uncertainty in instance unification. In *Proceedings of the Third ISWC Workshop on Uncertainty Reasoning for the Semantic Web*, volume 327 of *CEUR-WS*, 2007. Busan, Korea, November 12, 2007.

[13] C.-J. Liau. On rough terminological logics. In *Proceedings of the 4th International Workshop on Rough Sets, Fuzzy Sets and machine Discovery (RSFD'96)*, pages 47–54, 1996.

[14] C. Lutz, D. Toman, and F. Wolter. Conjunctive query answering in the description logic EL using a relational database system. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence IJCAI'09*. AAAI Press, 2009.

[15] B. Motik, B. C. Grau, I. Horrocks, Z. Wu, A. Fokoue, and C. Lutz. OWL 2 Web Ontology Language Profiles. W3C recommendation, W3C, 27 Oct. 2009. http://www.w3.org/TR/owl2-profiles/.

[16] B. Motik, P. F. Patel-Schneider, and B. C. Grau. OWL 2 web ontology language: Direct semantics. W3C recommendation, W3C, 27 October 2009 2009.

[17] B. Motik, R. Shearer, and I. Horrocks. Hypertableau Reasoning for Description Logics. *Journal of Artificial Intelligence Research*, 36:165–228, 2009.

[18] Z. Pawlak and A. Skowron. Rudiments of rough sets. *Information Sciences*, 177(1):3–27, 2007.

[19] S. Schlobach, M. Klein, and L. Peelen. Description logics with approximate definitions—precise modeling of vague concepts. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07)*, pages 557–562. AAAI Press, 2007.

[20] H. Viana, J. Alcantara, and A. T. Martins. Paraconsistent rough description logic. In *Proceedings of the 24th International Workshop on Description Logics (DL'11)*, 2011. Barcelona, July 13-16, 2011.

[21] A. Vitória, C. V. Damásio, and J. Maluszynski. From rough sets to rough knowledge bases. *Fundamenta Informaticae*, 57(2-4):215–246, 2003.

[22] Y. Yao and T. Lin. Generalization of rough sets using modal logic. *Intelligent Automation and Soft Computing*, 2(2):103–120, 1996.