On the feasibility of Description Logic knowledge bases with rough concepts and vague instances

C. Maria Keet

KRDB Research Centre, Free University of Bozen-Bolzano, Italy, keet@inf.unibz.it

Abstract. A usage scenario of bio-ontologies is hypothesis testing, such as finding relationships or new subconcepts in the data linked to the ontology. Whilst validating the hypothesis, such knowledge is uncertain or vague and the data is often incomplete, which DL knowledge bases do not take into account. In addition, it requires scalability with large amounts of data. To address these requirements, we take the $\mathcal{SROIQ}(D)$ and *DL-Lite* family of languages and their application infrastructures augmented with notions of rough sets. Although one can represent only little of rough concepts in *DL-Lite*, useful aspects can be dealt with in the mapping layer that links the concepts in the ontology to queries over the data source. We discuss the trade-offs and demonstrate validation of the theoretical assessment with the HGT application ontology about horizontal gene transfer and its 17GB database by taking advantage of the Ontology-Based Data Access framework. However, the prospects for *comprehensive and usable* rough DL knowledge bases are not good, and may require both sophisticated modularization and scientific workflows to achieve systematic use of rough ontologies.

1 Introduction

Various extensions of DLs and integration of DLs with other formalisms have been proposed, including to represent and reason over vague knowledge. To date, useful results have been obtained with fuzzy ontologies [1], but this is much less so for rough ontologies that aim to combine a standard DL with one of the formalisations of rough sets. In particular, [2–7] diverge in commitment as to which aspects of rough sets are included in the ontology language and the authors are concerned with the theory instead of demonstrating successful use of the rough DL in applications and ontology engineering. However, it has been noted within the Semantic Web context that scientist want to use ontologies together with data, such as hypothesizing that some subconcept exists and subsequently to validate this either in the laboratory or against the instances already represented in the knowledge base [8]. Such a hypothesised new concept is assumed to have a set-extension in the knowledge base and one would want to be able to match those instances with the right combination of object and data properties of the putative concept, i.e., taking a ‘guessed’ collection of attributes that is subsequently experimentally validated against the data and shown to be correct, or not; e.g., [9]. Such guessing includes dealing with incomplete or otherwise vague data, hence, for which some sort of *rough ontology* may be useful. Ideally, for all relevant individuals belonging to the putative concept, each value of the selected properties is distinct, but this may not be the case due to the limited data or insufficiency of the selected properties so that some individuals are indistinguishable from each other and therewith instantiating a rough concept. Despite the vagueness, it still can be useful in the ontology engineering process to include such a rough concept in the ontology. To support such usage of ontologies, one needs a language with which one can represent, at least, rough concepts as the intensional representation of the corresponding rough set and a way to (persistently) relate the data to the rough concepts. As it turns out, there is no perfect DL language, reasoner, and ontology development tool that does it all with respect to the semantics of rough sets, nor will there be if one adheres to the hard requirement of staying within the decidable fragment of FOL, let alone within the tractable zone. Some results can be obtained, however: in addition to representing most of rough sets' semantics with $\mathcal{SROIQ}$ using the TBox only, the linking to data and, moreover, ascertaining if a concept is really a rough concept can be achieved within the framework of Ontology-Based Data Access (OBDA) by exploiting the mapping layer [10]. While this, arguably, may not be perceived as a great outcome, it is possible (and the remainder of the issues can be passed on to an application layer with scientific workflows and refinements in the technologies). To demonstrate it is not merely theoretically possible to have rough concepts and vague instances in one's DL knowledge base, but that it is indeed practically possible, we take the use case about horizontal gene transfer with a hypothesized (rough) concept Promiscuous Bacterium, and demonstrate how this can be modelled more precisely in an OWL 2 DL ontology and deployed in an OBDA system using a $DL\text{-}Lite_{\mathcal{A}}$ ontology stored as an owl file so that the instances from the 17GB large HGT-DB database can be retrieved.

The remainder of the paper is structured as follows. We first introduce the basics of rough sets and discuss identification of rough concepts in Section 2. Trade-offs to include such roughness features in DLs will be discussed in Section 3. Results of the experimentation with rough concepts and with vague instances will be presented in Section 4, where we consider both the HGT ontology with the HGT-DB database and [7]'s septic patients. We close with conclusions in section 5.

2 Identifying rough concepts

To be able to have a correspondence of a rough set with a rough concept in an ontology and to represent its essential characteristics, we first outline the basics of rough sets following the standard "Pawlak rough set model" (see for a recent overview [11, 12]).

2.1 Rough sets

The Pawlak rough set model is depicted informally in Fig. 1 and formally, it is as follows. $I = (U, A)$ is called an *information system*, where U is a non-empty finite set of objects and A a finite non-empty set of attributes and 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 $P \subseteq A$, one can define the equivalence relation $\text{IND}(P)$ as

$$\text{IND}(P) = \{(x, y) \in U \times U \mid \forall a \in P, a(x) = a(y)\} \qquad (1)$$

$\textsc{ind}(P)$ 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*.

Given these basic notions, we can proceed to the definition of rough set. From the objects in universe U, we want to represent set X such that $X \subseteq U$ using the attribute set P where $P \subseteq A$. X may not be represented 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, respectively:

$$\underline{P}X = \{x \mid [x]_P \subseteq X\} \tag{2}$$

$$\overline{P}X = \{x \mid [x]_P \cap X \neq \emptyset\} \tag{3}$$

where $[x]_P$ denotes the equivalence classes of the p-indistinguishability relation. The *lower approximation* (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* 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*" [11], which is commonly 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 member of X nor that they are not in X; if $B_P X = \emptyset$ then X is, in fact, a crisp set with respect to P and when $B_P X \neq \emptyset$ then X is rough w.r.t. P.

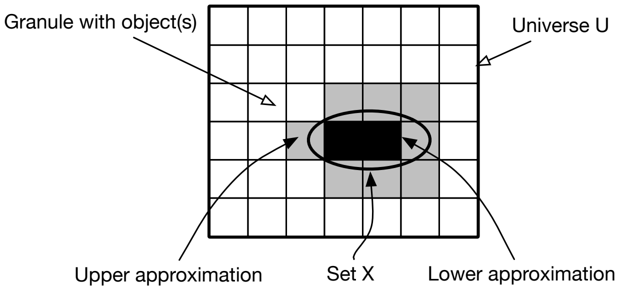

Fig. 1. A rough set and associated notions (Source: based on [11]).

The *accuracy of approximation* provides a measure of how closely the rough set is approximating the target set with respect to the attributes in P. There are several of such measures, denoted with $\alpha_P X$, for instance $\alpha_P X = \frac{|\underline{P}X|}{|\overline{P}X|}$ and $\alpha_P X = 1 - \frac{|B_P X|}{|U|}$. Clearly, if $\alpha_P X = 1$, then the boundary region $B_P X$ is empty and thus X is crisp.

Useful for subsequent sections is also the following property of approximations:

$$\underline{P}X \subseteq X \subseteq \overline{P}X \tag{4}$$

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., $\textsc{Core}$ = $\textsc{Red}_1 \cap \ldots \cap \textsc{Red}_n$. That is, those attributes that are in P but not in $\textsc{Red}$ are superfluous with respect to the partitioning. On the other hand, no attribute in $\textsc{Core}$ can be removed without destroying the equivalence structure (it is possible that $\textsc{Core}$ is an empty set).

2.2 Some ontological considerations

As a first step toward rough ontologies, it would be a severe under-usage of DL knowledge bases if one only were to copy Pawlak's 'information system' essentials, because

1. In a logic-based (formal) ontology we have more constructors and possible constraints at our disposal, most notably a set of roles, $\mathcal{R}$, over objects and universal and existential quantification;
2. There is more flexibility on how to represent 'attributes' of a concept $C \in \mathcal{C}$: either with one or more roles $R \in \mathcal{R}$ (i.e., object properties in OWL) or value attributions $D \in \mathcal{D}$ (i.e., data properties in OWL), or both;
3. We need a complete and appropriate model-theoretic semantics for $\underline{C}$ and $\overline{C}$, and, as counterpart of the rough set, a rough concept, which we denote with "$\wr C$" for presentation convenience to clearly distinguish it from a crisp concept;
4. Given that attributes are used to compute $\underline{C}$ and $\overline{C}$, then those attributes must be represented in the ontology, and with $\wr C$ a tuple of the former two, then also it must have the attributes recorded in the ontology.

Concerning item 3, the semantics of the approximations is fairly straightforward, with E denoting the reflexive, symmetric and transitive indistinguishability (equivalence) relation:

$$\underline{C} = \{x \mid \forall y : (x, y) \in E \rightarrow y \in C\} \tag{5}$$

$$\overline{C} = \{x \mid \exists y : (x, y) \in E \wedge y \in C\} \tag{6}$$

Then there is rough sets' tuple notation, $X = \langle \underline{X}, \overline{X} \rangle$, for which we may have an analougous one for concepts, $\wr C = \langle \underline{C}, \overline{C} \rangle$. For $\wr C$, there are two issues: the notational distinction between a crisp (C) and a rough ($\wr C$) concept, and the tuple notation. Regarding the first issue, there are two ontological commitments one can take regarding the sets—either X is a special type of rough set where $\alpha = 1$ or a rough set is a special type of a crisp set because it is defined by the two crisp sets $\underline{X}$ and $\overline{X}$—and, subsequently, if a 'rough ontology' consists of only rough concepts or may contain both rough concepts and crisp concepts. Because rough sets are defined in terms of crisp sets, and, correspondingly, rough concepts in terms of a combination of two crisp concepts, this means that the crisp set and concepts are the 'primitive' ones and that we end up with a rough ontology that has both rough and crisp concepts to be able to have rough concepts properly defined in an ontology. For this reason, we maintain the, thus far, syntactic distinction between a crisp concept C and a rough concept $\wr C$. Regarding the second point, and, in fact, the semantics of $\wr C$, using a tuple notation is not ideal for discussing ontological commitments of rough sets and rough concepts and so it is useful to flatten it out. One can commit to the subsumption relation between the sets as in (4) and their corresponding concepts as pursued by [5, 7] or take a more flexible approach that subsumes the former by introducing two binary relationships, $lapr$ and $uapr$, to relate *any* rough concept and its associated approximations, which are typed as follows:

$$\forall \phi, \psi . lapr(\phi, \psi) \rightarrow \wr C(\phi) \wedge \underline{C}(\psi) \tag{7}$$

$$\forall \phi, \psi . uapr(\phi, \psi) \rightarrow \wr C(\phi) \wedge \overline{C}(\psi) \tag{8}$$

Observe that here we are quantifying over *sets*, not objects that are member of the respective sets; i.e., we make explicit the knowledge about the three sets and how they relate, not about the instances in those sets. With these relations we can make explicit that $\wr C$ is identified by the combination of its $\underline{C}$ and $\overline{C}$, which is achieved by the following set of constraints:

$$\begin{aligned}
&\forall \phi . \wr C(\phi) \rightarrow \exists \psi . lapr(\phi, \psi), \\
&\forall \phi . \wr C(\phi) \rightarrow \exists \psi . uapr(\phi, \psi), \\
&\forall \phi, \psi, \varphi . lapr(\phi, \psi) \wedge lapr(\phi, \varphi) \rightarrow \psi = \varphi, \\
&\forall \phi, \psi, \varphi . uapr(\phi, \psi) \wedge uapr(\phi, \varphi) \rightarrow \psi = \varphi, \\
&\forall \phi_1, \phi_2, \psi_1, \psi_2 . lapr(\phi_1, \psi_1) \wedge uapr(\phi_1, \psi_2) \wedge \\
&\quad lapr(\phi_2, \psi_1) \wedge uapr(\phi_2, \psi_2) \rightarrow \phi_1 = \phi_2 .
\end{aligned} \tag{9}$$

The axioms in (9) say that for each rough concept, there must be exactly one lower approximation and one upper approximation and for each combination of lower and upper approximation, there is one rough concept, i.e., if either one of the approximations differ, we have a different rough concept.

Last, because a partitioning of the universe of objects is done by means of selecting a specific subset P of A of rough sets' information system, we have in the DL notion of ontology that the set of 'attributes' amounts to $\mathcal{R} \cup \mathcal{D}$. Moreover, one has to impose at the knowledge layer that those attributes P taken from $\mathcal{R} \cup \mathcal{D}$ must be represented in the ontology with $\wr C$ as its domain so as to represent explicitly and persistently which properties were used to obtain the rough set as extension of $\wr C$.

Overall, we thus have a more precise notion of $\wr C$ cf. the tuple notation in [5], use both $\mathcal{R}$ and $\mathcal{D}$ for the 'attributes' (properties) of the concepts (cf. $\mathcal{R}$ only in [4, 7]), include the properties of the indistinguishability/equivalence relation (cf. their omission in [6] or using the properties of the similarity relation [2]), and adhere to proper declaration of $\underline{C}$, $\overline{C}$, and $\wr C$ in that they all have the same collection of properties from $\mathcal{R} \cup \mathcal{D}$ (cf. giving the 'approximations' different sets of attributes in [7]).

3 Considerations regarding rough DL knowledge bases

The previous section introduced two essential aspects for a rough ontology language: the necessity to represent the indistinguishability relation E and declare it reflexive, symmetric, and transitive, and the identity of a rough concept by its lower and upper approximation by means of identification constraints involving DL roles. Currently, there is no DL language with corresponding complexity results that has both features.

On the one hand, one could decide to invent a new language that includes both features and that is hopefully still tractable in the light of abundant data. However, if one were to be faithful to (7-9), then a second order logic is required, which is out of scope. Alternatively, identification constraints (**id**s) have to be added in the ontology for each rough concept (perhaps guided with an outside-the-langauge ontology design pattern), hence the requirement to have the more common **id** constraint in the language. On the other hand, one can decide to push the envelope of extant languages and tools and make concessions. From a KR perspective, the former may be more interesting, but with an eye on applicability and demands from the most active user-base of ontologies—the life sciences and health care—it is worthwhile to push extant languages and its related tools as far as possible to gain better insight if development of a new language and corresponding tools are worth the effort. Give the extant languages, $\mathcal{SROIQ}(D)$ [13] suffices for representing E, but not **id** and it does not behave well in the light of large ABoxes, whereas the languages in the $DL\text{-}lite$ family [10] are well-suited to handle large ABoxes, but then we cannot represent E's relational properties and the **id** can be represented only in $DL\text{-}Lite_{\mathcal{A},id}$. Some other DL languages, such as $\mathcal{DLR}_{\mathit{ifd}}$ and $\mathcal{DLR}_{\mu}$, also have either one or the other feature, but not both.

For *practical* reasons, we narrow down the DL knowledge base further to the DL-based OWL species, because they are W3C standardised languages, there are ontology development tools for them, they have automated reasoners, and they are the DL of choice among the bio-ontologists. If we represent the reflexivity, symmetry and transitivity of E, then we are confined to the new OWL 2 DL, for this is the only one where one can assert all three object properties [14, 15]. For $\wr C$, there are two principal options: either define its semantics outside the language, or declare a "RoughC" in the ontology and let all rough concepts also be subsumed by it. In the latter option and considering the ontology languages and tools such as Protégé and Racer, we cannot represent the identification constraint anyway (nor the tuple notation $\wr C = \langle \underline{C}, \overline{C} \rangle$ proposed by [5]), and for the former option the applications would have to be adjusted to include a check if the rough concepts are declared correctly. Moreover, one should ask oneself what can be gained from including $\underline{C}$ and $\overline{C}$ in the ontology, besides deducing $\underline{C} \sqsubseteq C \sqsubseteq \overline{C}$ based on the declared knowledge in the TBox (thanks to (5) and (6)). Jiang and co-authors identify the specific TBox reasoning services for their $\mathcal{RDL}_{AC}$ as definitely satisfiable, possibly satisfiable, and rough subsumption [5]. However, considering rough sets' usage, it is the interplay with the actual instances that is crucial: after all, it is only based on the fact that, *given a non-empty ABox*, the boundary region is not empty that makes a concept a rough concept, and if we do not even test it against the actual instances in the knowledge base, then there is no point in bothering oneself to include a merely hypothetical rough concept in the ontology that cannot be examined either way if it really is a rough concept.

Thus, another hurdle is the data, which can be loaded into the ABox proper or stored and dealt with in secondary storage. Considering the most widely used ontology development tool Protégé, it loads the ABox in main memory, which is doable for small data sets but not for the medium to large size biological databases that easily exceed several GB. Setting aside supercomputers and the obstacle to wait a while on a query answer, this, then, forces one to take the second option of secondary storage, which, in turn and at the time of writing, locks one into *DL-Lite* (and for the bio-ontologist, OWL 2 QL) that can represent even less of rough set's semantics (and of the subject domain) than OWL 2 DL. With the latter option, and, realistically, the Ontology-Based Data Access framework with QuOnto [10], the lack of expressiveness of the language can be counterbalanced by putting some of the subject domain semantics in the mapping layer. This is not ideal because it is not as maintainable as when it would be represented in the ontology, and it is not transparent for the domain expert who ideally should query just the ontology and not bother with the knowledge squeezed into the mapping layer, but we can get the data out of the database and have our rough concepts.

4 Experimentation with a rough ontology and vague instances

Given these trade-offs, we will demonstrate how one can have either an ontology with rough concepts represented fairly comprehensively regarding the semantics (in Experiment 1) or have it with more limited semantics but linked to the data and be able to perform the actual hypothesis testing against the data (Experiment 2) using the HGT as use case. To be fair to the latest technologies for expressive DLs, we also experimented with a more expressive ontology than the HGT ontology and then using much less data, by revisiting the definitions of septic of [7] and data of just 17 patients. Additional files (ontologies, mappings, queries, and data) are available online as supplementary material at `http://obda.inf.unibz.it/obdahgtdb/obdahgtdb.html`. The results will be discussed in Section 4.2.

4.1 Results

The background for Experiment 1 and 2 is as follows. A geneticist has an idea about what a "promiscuous bacterium" is because some bacteria transfer and receive much more genes from other bacteria than others do. It is not fully understood who they are and why this is the case, hence, the first step is to analyse the data—in casu, stored in the 17GB HGT-DB database—using properties that indicate a certain promiscuity so as to find bacteria with comparatively many anomalous (foreign) DNA in their chromosome.

Experiment 1 (Promiscuous bacteria in OWL 2 DL) We specify a first attempt for representing the promiscuous bacterium ($PromBact$) as a subtype of $Bact$erium in the HGT ontology with an additional object- and a data property, so that it must have more than 5 so-called flexible hgt-gene clusters ($FlexCl$, which are sets of adjacent or nearby genes that are horizontally transferred) and the percentage of genes on the chromosome that are predicted to be horizontally acquired, $hgtPerctg$, as > 10:

$$PromBact \equiv Bact \sqcap \exists\, hgtPerctg.real_{>10} \sqcap \geq 6\ hasHGTCluster.FlexCl \quad (10)$$

In addition, we can add the assertions regarding the equivalence relation (relational properties omitted for brevity) and that $PromBact$ has exactly one lower and one upper approximation, $PromBactLapr$ and $PromBactUapr$, as follows:

$$PromBact \sqsubseteq =1\ lapr.PromBactLapr \tag{11}$$

$$PromBact \sqsubseteq =1\ uapr.PromBactUapr \tag{12}$$

$$PromBactLapr \equiv \forall E.PromBact \tag{13}$$

$$PromBactUapr \equiv \exists E.PromBact \tag{14}$$

Running ahead of the data we retrieve with OBDA, $PromBact$ is indeed a rough concept, so we also have specified a refinement, $PromBact'$ to investigate if we now have included enough properties to have an empty boundary, hence a crisp concept:

$$\begin{aligned} PromBact' \equiv PromBact\ \sqcap \exists\ hgtPerctg.real_{>10} \sqcap \\ \geq 11\ hasHGTCluster.FlexCl\ \sqcap nrHGTgenes.integer_{>150} \end{aligned} \tag{15}$$

Querying or instance classification with this OWL 2 DL version and the HGT data is currently not feasible. ♢

Experiment 2 (Promiscuous bacteria in OBDA) As in Experiment 1, our first attempt is to represent $PromBact$ in $DL\text{-}Lite_{\mathcal{A}}$ (roughly OWL 2 QL), where we do not have existential quantification in the subclass position, no cardinality restrictions, limited object property assertions, no class equivalence, and no data property restrictions. To not have the intended meaning of $PromBact$ as in (10) all over the place, we chose to put it in the OBDA mapping layer; that is, we have $PromBact \sqsubseteq Bact$ in the $DL\text{-}Lite_{\mathcal{A}}$ ontology, and then make a mapping between $PromBact$ in the ontology and a SQL query over the relational database (for technical details about the OBDA framework used, the reader is referred to [10]). The head of the mapping is:

```
PromBact(getPromBact($abbrev,$ccount,$percentage))
```

and the body, i.e. the SQL query over the database where the WHERE clause has the set of interesting properties for $PromBact$ (which were modelled as object and data properties in the TBox in the previous experiment):

```
SELECT organisme.abbrev, ccount, organisme.percentage
   FROM ( SELECT idorganisme, COUNT(distinct cstart)
      as ccount FROM COMCLUSTG2 GROUP BY idorganisme
     ) flexcount, organisme
WHERE organisme.abbrev = flexcount.idorganisme AND
      organisme.percentage > 10 AND flexcount.ccount > 5
```

Querying the database through the ontology with a SPARQL query using the OBDA Plugin for Protégé and answered using DIG-QuOnto, 98 objects are retrieved where *Dehalococcoides CBDB1* and *Thermotoga maritima* are truly indistinguishable bacteria, i.e. they have the same values for all the selected and constrained attributes, and a few others are very close to being so, such as *Pelodictyon luteolum DSM273* and *Synechocystis PCC6803* who have both 6 clusters and 10.1% and 10.2%, respectively, (which, practically, still lie within the error-margin of genomics data and its statistics); see online material for details. Hence, $PromBact$ is actually a rough concept.

To improve the accuracy and examine if we can turn a subconcept of $PromBact$ into a crisp concept, a *new* data property—$NrOfHGTgenes$ with integer values, set to >150—is added and the second attribute set at >10 gene clusters, which thus *revises* the assumption of what a promiscuous bacterium really is, i.e., we have $PromBact'$ in the ontology such that $PromBact' \sqsubseteq PromBact$. The head of the mapping is:

```
PromBactPrime(getPromBactPrime($abbrev,$ccount,$percentage,$hgt))
```

and the body:

```
SELECT organisme.abbrev,ccount,organisme.percentage,organisme.hgt
  FROM  ...
WHERE organisme.abbrev = flexcount.idorganisme AND
      organisme.percentage > 10 AND flexcount.ccount > 10 AND
      organisme.hgt > 150
```

The query answer has only 89 objects and this change even eliminates the boundary region, hence $PromBact'$ is a *crisp* concept with respect to the database. ♢

Experiment 3 (Revisiting septic patients) Patients may be septic or are certainly septic, according to the so-called *Bone criteria* and Bone criteria together with three out of another five criteria, respectively. For instance, the Bone criteria are (from [7]):

- *Has infection;*
- *At least two out of four criteria of the Systemic Inflammatory Response Syndrome:*
  - *temperature* $> 38°C$ *OR temperature* $< 36°C$*;*
  - *respiratory rate* > 20 *breaths/minute OR PaCO*$_2$ < 32 *mmHg;*
  - *heart rate* > 90 *beats/minute;*
  - *leukocyte count* < 4000 *mm*^{3} *OR leukocyte count* > 12000 *mm*^{3}*;*
- *Organ dysfunction, hypoperfusion, or hypotension.*

The respective encodings in Protégé 4.0 and RacerPro 2.0 preview are available online as supplementary material, as well as data of 17 'patients' such that the boundary region of each concept is not empty. The experiments were carried out on a Macbook Pro with Mac OS X v 10.5.8 with 2.93 GHz Intel core 2 Duo and 4 GB memory. Protégé 4.0 with Pellet 2.0 did not work at all. Protégé 4.0 with FaCT++ works well with a few dummy concepts and a few instances, but the esoteric definitions for septic appeared to be more challenging: it crashed with an encoding including the indistinguishability relation E and (with or without E), upon saving and reopening the owl file it had reordered the braces in the definition in such a way as to change its meaning so that it does not classify all 17 individuals correctly. These observations may be due to the fact that the software used is still in the early stages. RacerPro 2.0 preview never crashed during exerimentation and did return the correct classifications within about 2 hours. While the latter is an encouraging result because it works with the real definitions and a small data set, the automated reasoning clearly does not scale to [7]'s thousands of patients. (The authors did not respond on a request for details of their experimental set-up.) ♢

4.2 Discussion

While a rough ontology such as the amended HGT ontology in OWL 2 DL can provide a better way of representing the declarative knowledge of putative and actual rough concepts, it is only with the less expressive *DL-Lite*-based OBDA system that it could be experimentally validated against the data. The ontologies and OBDA provide a means to represent the steps of successive de-vaguening during experimentation, they make the selected properties explicit, and, if desired, one can keep both $\wr PromBact$ and $PromBact'$ in the ontologies without generating inconsistencies.

However, TBox rough subsumption and possible and definite satisfiability reasoning might be useful during engineering of rough ontologies. To improve outcomes for the expressive ontology setting, one could split up the database and import into the ABox all the data of only one organism at a time, do the instance classification, export the results, merge the results after each classification step, and then manually assess them. However, there are currently about 500 organisms in the database (which are soon to be extended to about 1000) and, ideally, this should not be done with one-off scripting. Alternatively, one may be able to design sophisticated modularization of both the ontology and the data(base) so as to execute the reasoning only on small sections of the ontology and database, in the direction of, e.g., [16, 17].

Although a rough DL knowledge base works as proof-of-concept, the procedure to carry it out is not perceived to be an ideal one. One might be able to turn into a feature the cumbersome interaction between the more precise representation of rough concepts in OWL 2 DL and the linking to data with OWL 2 QL (or a similar tractable language) by upgrading it to a named *scientific workflow*. This guides the developer to carry out in a structured, traceable, and repeatable manner the tasks to (*i*) develop a basic ontology in OWL 2 QL or $DL\text{-}Lite_{\mathcal{A}}$, (*ii*) get the database, (*iii*) set up the OBDA system, (*iv*) declare the mappings between the concepts and roles in the ontology and SQL queries over the database, (*v*) find all rough concepts with respect to the data and add them to the ontology, (*vi*) migrate this ontology to OWL 2 DL, (*vii*) add the semantics from the WHERE clause in the SQL query of the mapping layer as object and data properties in the ontology, (*viii*) add upper and lower approximations of each rough concept, (*ix*) add the equivalence relation with its properties, (*x*) add the axioms relating the approximations to the rough concepts and vice versa, and (*xi*) when the rough reasoning services are implemented, run the reasoner with the enhanced ontology. It will also be useful to go in the reverse direction in the light of updates to the database and in case the ontology was inconsistent or a had an unsatisfiable concept.

5 Conclusions

Extension of standard Description Logics knowledge bases with the essential notions of rough sets revealed both theoretical and practical challenges. Given rough sets' semantics, there is no, nor will there be, a DL that represents all essential aspects precisely, although expressive languages, such as $\mathcal{SROIQ}(D)$, come close and some tools, such as RacerPro, can handle complex rough concept descriptions with a small amount of data. On the other hand, it is the interaction with large amounts of data that makes any extension with roughness interesting and useful. This can be addressed with a tractable Ontology-Based Data Access framework by exploiting the mapping layer that links the concepts in the ontology over queries to the database. To validate the theoretical assessment, we have experimented with rough concepts and vague instances using the HGT case study and the recurring example of septic patients. The experimentation showed it is possible to have rough knowledge bases. However, more work in the direction of streamlining the rather elaborate procedure into a scientific workflow or developing implementations of sophisticated ontology and data modularization, or both, is advisable in order to achieve a platform for hypothesis-driven usage of rough ontologies that will reap the greatest benefits to meet the users' requirements.

Acknowledgements I thank Umberto Straccia, Ferdinando Bobillo, and Mariano Rodríguez-Muro for feedback during the experimentation.

References

1. Lukasiewicz, T., Straccia, U.: Managing uncertainty and vagueness in description logics for the semantic web. Journal of Web Semantics **6**(4) (2008) 291–308
2. Bobillo, F., Straccia, U.: Supporting fuzzy rough sets in fuzzy description logics. In: Proc. of ECSQARU'09. Volume 5590 of LNCS., Springer (2009) 676–687
3. Fanizzi, N., D'Amato, C., Esposito, F., Lukasiewicz, T.: Representing uncertain concepts in rough description logics via contextual indiscernibility relations. In: Proc. of URSW'08. Volume 423 of CEUR-WS. (2008)
4. Ishizu, S., Gehrmann, A., Nagai, Y., Inukai, Y.: Rough ontology: extension of ontologies by rough sets. In Smith, M.J., Salvendy, G., eds.: Proceedings of Human Interface and the Management of Information. Volume 4557 of LNCS., Springer (2007) 456–462
5. Jiang, Y., Wang, J., Tang, S., Xiao, B.: Reasoning with rough description logics: An approximate concepts approach. Information Sciences **179** (2009) 600–612
6. Liau, C.J.: On rought terminological logics. In: Proceedings of the 4th International Workshop on Rough Sets, Fuzzy Sets and machine Discovery (RSFD'96). (1996) 47–54
7. Schlobach, S., Klein, M., Peelen, L.: Description logics with approximate definitions—precise modeling of vague concepts. In: Proc. of IJCAI'07, AAAI Press (2007) 557–562
8. Keet, C.M., Roos, M., Marshall, M.S.: A survey of requirements for automated reasoning services for bio-ontologies in OWL. In: Proc. of OWLED'07. Volume 258 of CEUR-WS. (2007) 6-7 June 2007, Innsbruck, Austria.
9. Marshall, M.S., Post, L., Roos, M., Breit, T.M.: Using semantic web tools to integrate experimental measurement data on our own terms. In: Proc. of KSinBIT'06. Volume 4277 of LNCS., Springer (2006) 679–688
10. Calvanese, D., et al.: Ontologies and databases: The DL-Lite approach. In: Semantic Technologies for Informations Systems. Volume 5689 of LNCS., Springer (2009) 255–356
11. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences **177**(1) (2007) 3–27
12. Pawlak, Z., Skowron, A.: Rough sets: Some extensions. Information Sciences **177**(1) (2007) 28–40
13. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible $\mathcal{SROIQ}$. Proc. of KR'06 (2006) 452–457
14. Motik, B., Patel-Schneider, P.F., Parsia, B.: OWL 2 web ontology language structural specification and functional-style syntax. W3c recommendation, W3C (27 Oct. 2009) http://www.w3.org/TR/owl2-syntax/.
15. Motik, B., Grau, B.C., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C.: OWL 2 Web Ontology Language Profiles. W3c recommendation, W3C (27 Oct. 2009) http://www.w3.org/TR/owl2-profiles/.
16. Lutz, C., Toman, D., Wolter, F.: Conjunctive query answering in the description logic el using a relational database system. In: Proc. of IJCAI'09, AAAI Press (2009)
17. Baader, F., Bienvenu, M., Lutz, C., Wolter, F.: Query and predicate emptiness in description logics. In: Proc. of KR'10. (2010)