

Context-dependent Reasoning for the Semantic Web

Neli P. ZLATAREVA

Department of Computer Science,
Central Connecticut State University
1615 Stanley Street, New Britain, CT 06050, USA

ABSTRACT

Ontologies are the backbone of the emerging Semantic Web, which is envisioned to dramatically improve current web services by extending them with intelligent capabilities such as reasoning and context-awareness. They define a shared vocabulary of common domains accessible to both, humans and computers, and support various types of information management including storage and processing of data. Current ontology languages, which are designed to be decidable to allow for automatic data processing, target simple typed ontologies that are completely and consistently specified. As the size of ontologies and the complexity of web applications grow, the need for more flexible representation and reasoning schemes emerges. This article presents a logical framework utilizing context-dependent rules which are intended to support not fully and/or precisely specified ontologies. A hypothetical application scenario is described to illustrate the type of ontologies targeted, and the type of queries that the presented logical framework is intended to address.

Keywords: Semantic Web Technologies, Web Ontologies, Uncertain Reasoning, Production Rules, Rule-based Web Services.

1. INTRODUCTION

World Wide Web is undergoing a remarkable transformation intended to dramatically improve current web services by extending them with intelligent capabilities such as reasoning and context-awareness. This will make the new Semantic Web [1] equally accessible to humans and computers, thus making it possible for web agents to not only search and retrieve information, but also to make decisions and act on consumer behalf. Consider, for example, one of the largest web services, business-to-consumer e-commerce. The way it currently works is the following. To buy a product, a consumer visits several online shops to compare (manually) their prices, special offers, etc. with no guarantee that the “best deal” will be found because of the limited search involved. Compare this ad-hoc approach to the following scenario. The customer enters desired product specifications and leaves the rest of the search to his computer which autonomously navigates through online retailers offering the product, collects and evaluates offers, and returns one or several best offers to the customer for final decision.

To implement this scenario, the following two problems must be solved: (i) the domain navigated by the computer must be represented in a machine-understandable form, and (ii) an inference engine able to process not fully and/or precisely specified queries in order to adequately match customer requirements to product descriptions found on the web, must be in place.

In the past decade, a lot of research coordinated by the World Wide Web Consortium (W3C) [2] was conducted to address the first problem. Most of this research however was carried out from representation- or functionality-centered perspective, and resulted in developing ontology languages with limited representation and inference capabilities. Among best known ones are XML [3], RDF [4], RDFS [5], DAML+OIL [6], and OWL [7]. XML was designed as universal metalanguage for defining markup with no reference to data semantics. RDF, which is built upon the XML syntax, is basically a domain independent data model intended to represent simple typed ontologies defined by binary ground predicates. RDFS is an extension of RDF to allow for representing subclass and subproperty relationships, but it imposes a local scope of properties and is unable to define disjointness or Boolean combination of classes. OWL, which was recently recommended by W3C as a standard web ontology language, provides more expressive representation, but its inference capabilities are limited to satisfiability, subsumption, equivalence and disjointness. Composition of properties cannot be expressed in OWL, thus making it inadequate to address the second problem above.

As the size of ontologies and the complexity of web applications grow, uncertainty, incompleteness and inconsistency are becoming common properties of ontological knowledge. Rules were shown to be very effective in processing such knowledge, and future Semantic Web services are expected to depend heavily on them. The RuleML initiative [8] is the earliest effort to define a normalized markup for representing and exchanging rules on the Semantic Web. Although RuleML is currently limited to Horn rules, significant research efforts are underway to extend it with more flexible representation and reasoning capabilities [9, 10, 11, 12]. Most of this research is based on non-classical logics (probabilistic, fuzzy, possibilistic, etc.) which are primarily concerned with uncertainty and incompleteness of knowledge, assuming its consistency. Semantic Web, being an open and highly dynamic environment, will inevitably contain inconsistencies. The importance of their adequate processing in order to maintain the

validity of ontological knowledge has been widely acknowledged by the Semantic Web community, but little work has been done so far to develop techniques and tools for reasoning with inconsistent knowledge ([11] and [12] are notable exceptions).

In [13], we have outlined one reasoning technique utilizing *context-dependent rules*, which we advocated was a good candidate for Semantic Web applications because of its expressiveness, adaptability and computational efficiency. In this article, we describe a hypothetical application scenario to further discuss the importance and role of context-dependent reasoning for ontologies that are not fully and/or precisely defined. To support such ontologies, the inference procedure should be able to maintain incomplete, uncertain, and even inconsistent specifications, preserving at the same time the validity and meaningfulness of derived knowledge. Because “meaningfulness” in this case can only be evaluated with respect to a particular context, the inference procedure should be able to identify, maintain and interpret that context when addressing a specific query. In this article, we show how this can be done for one type of imprecisely specified domains. An example of such a domain is presented in Section 3. In section 4, we discuss a variation of the logical framework from [13] and in Section 5 we illustrate its suitability for representing and handling our example domain. But first, in section 2 we briefly discuss some of the limitations of current RuleML to justify the need for more expressive rule representation format.

2. RuleML LIMITATIONS: A PRACTICAL WAY TO ADDRESS THEM

RuleML is an XML-based language for representing and exchanging different kinds of Horn rules (derivation rules, reaction rules, integrity constraints) on the Semantic Web. As we show next, its limited expressiveness is not sufficient to represent some common situations that may arise in various web applications. Here is one such application (example adapted from [14]).

“Carlos is looking for an apartment of at least 45 sq m with at least 2 bedrooms. Carlos is willing to pay \$300 for a centrally located 45 sq m apartment, but \$250 for a similar one in the suburbs. He will pay extra for a larger apartment or an apartment with a garden. Carlos does not want to pay more than \$400, but if the apartment is centrally located, offers a swimming pool, and have other desired features, he may consider a higher rent. If the apartment is on the third floor or higher, the building must have an elevator. Pets must be allowed, because Carlos cannot leave his dog behind. Given a choice, the price will be Carlos’ first priority, but amenities (swimming pool, garden, location) will also play a role in Carlos’ final decision.”

Some of Carlos’ requirements are quite vague (location, swimming pool, garden), while others are very specific and firm (pet friendliness, number of bedrooms, size). Firm requirements can be easily expressed in RuleML. For example, “Carlos is looking for an apartment of at least 45 sq m” can be represented as the following derivation rule:

```
<Implies>
  <Body>
    <And>
      <Atom>
        <Rel>size</Rel>
        <Var>Apartment</Var>
        <Var>X</Var>
      </Atom>
      <Atom>
        <Rel>GreaterThan</Rel>
        <Var>X</Var>
        <Ind>45 sq m</Ind>
      </Atom>
    </And>
  </Body>
  <Head>
    <Atom>
      <Rel>consider</Rel>
      <Var>Apartment</Var>
    </Atom>
  </Head>
</Implies>
```

Representing statements, such as “an apartment may be OK with or without a garden”, however, is not straightforward without enforcing some change in the meaning of the statement. What we need in order to adequately match such “relaxed” statements to rule premises, is to allow for “relaxed” premises as well. For example, to say that Carlos will consider an apartment which costs less than \$400 despite of its location or availability of a garden, we need a rule of the following type:

```
If      A: size > 45 and bedroom > 1 and pets-allowed
IN SPITE garden or location = Central
Then   A: consider
```

According to this rule, an apartment A will be considered if Carlos’ firm requirements (size > 45, bedroom > 1, pets-allowed) are met. But if the apartment is centrally located and/or has a garden, it will be “even better”.

To represent and process such rules, we need more than a simple Horn-style syntax and classical forward or backward chaining. Non-monotonic logics will also not work in this case, because their non-monotonic premises have a completely different semantics – they serve as exceptions to the rule’s conclusion. However, various possibilistic logics (probabilistic, fuzzy, etc.) intended to handle uncertain knowledge do have means to represent rules with “relaxed” (uncertain) premises by associating a “degree of certainty” with each premise and employing “combining functions” to maintain the uncertainty during the reasoning process. There are two main problems with these logics, though: (i) it is difficult to provide a reasonable interpretation for numerical values of certainty, and (ii) combining functions are application-dependent. Truth maintenance logics [15, 16] address this difficulty by providing explicit justifications for derived statements instead of numerical values thus defining the context within which the meaning of the statement must be interpreted.

Next, we discuss an extension to the basic RuleML format intended to accommodate “relaxed” premises in order to allow

for efficient processing and interpretation of not completely or precisely specified ontologies. The motivation example below illustrates the type of ontologies targeted and the type of queries that the presented framework is intended to support.

3. MOTIVATION EXAMPLE

Consider two ontologies from the “University” domain shown on Figures 1 and 2. Assume they represent core curriculums of two different graduate programs. Both ontologies utilize concepts of the following two types:

- **Generic concepts**, which are common for the entire domain. These are marked as UGj, (undergraduate-level courses).
- **Ontology-specific concepts**, such as GRni (graduate-level courses) which are defined only in the ontology they belong to.

The domain semantics suggests two different types of relations between concepts, which are expressed as lines and dotted lines on Figures 1 and 2. The meaning of these relations is given next.

Ontology A:

- 1A: GR1A requires UG1 and UG2, but UG3 and UG4 are recommended.
- 2A: GR2A requires UG6 and UG7, but UG8 is recommended.
- 3A: GR4A requires GR1A and UG8.
- 4A: GR3A requires UG4 and GR1A, but UG9 is recommended.
- 5A: GR2A requires GR4A.
- 6A: GR4A requires GR3A and UG6.
- 7A: GR5A requires GR4A.
- 8A: GR6A requires GR3A, but UG10 is recommended.

Ontology B:

- 1B: GR1B requires UG1 and UG2, but UG3 is recommended.
- 2B: GR2B requires UG9, but UG7 is recommended.
- 3B: GR3B requires GR2B and UG4, but UG6 is recommended.
- 4B: GR3B requires GR1B and GR2B.
- 5B: GR4B requires GR2B and UG9, but UG8 is recommended.
- 6B: GR3B requires GR1B and UG6.
- 7B: GR5B requires GR3B and GR4B.
- 8B: GR6B requires GR2B and GR3B.

Notice that the relations marked with dotted lines are not enforced. For example, according to relation 1A to take GR1A a student must have UG1 and UG2, but she is not required to have UG3 or UG4. What if a student does have UG3 or UG4? Is she going to be more successful in GR1A? How different are UG3 and UG4 from UG1 and UG2? What if the student does not have UG1, but instead has UG3 and UG4?

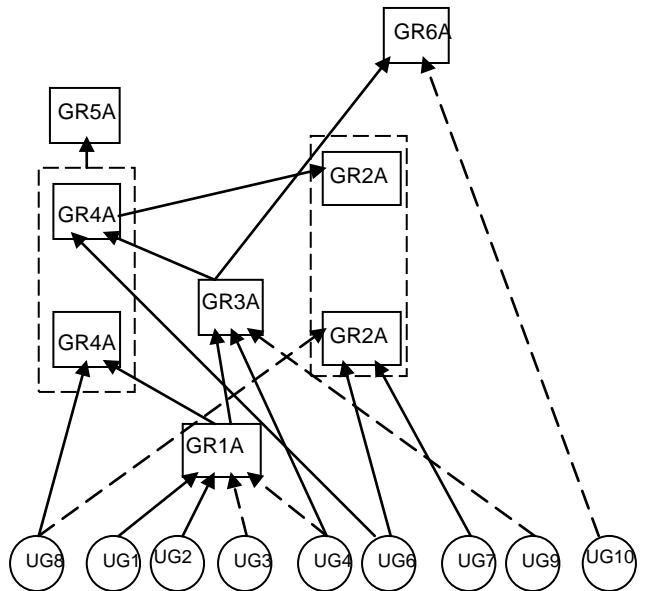


Figure 1

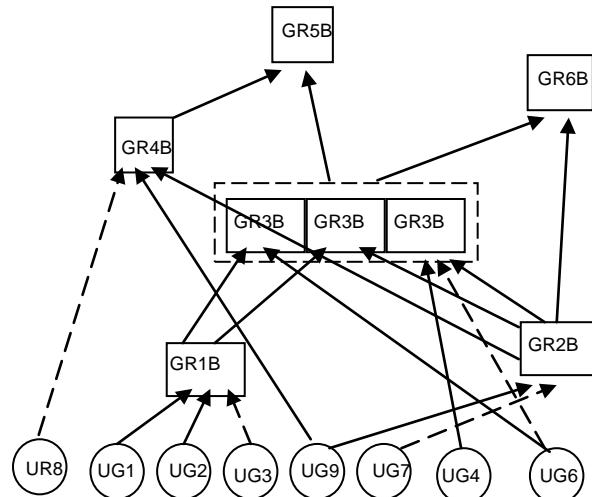


Figure 2

Although the answers to these questions depend on the semantics of the domain, a logical framework able to represent and support relations with different degrees of significance would be able to answer questions like “Is this program a good match for me?” and “Given my specific background, which programs will allow me to graduate faster?” Because the reasoning involved in addressing such queries is non-monotonic, the validity of derived concepts should be evaluated with respect to the “context” in which derivations take place.

Consider, for example, two students Anna and Peter who are looking for a graduate program which would allow them to graduate in the shortest time and/or matches their undergraduate background the best. Assume that Peter has completed UG1, UG2, UG3, UG4, UG6, and UG9 (out of 10 standard courses that his undergraduate major suggests), and Anna has

completed UG1, UG2, UG3, UG6, UG7, and UG8. Given the two example ontologies, Peter will be better off choosing program B, if he wants to complete the core courses in the shortest time. However, if he is looking for a program that fits his undergraduate experience the best, then program A may be a better choice (see Section 5). For Ana, program A might be a better choice, because although it requires UG4 which Ana does not have, she will be able to complete it in three semesters, while program B (which also requires a remedial course, UG9) will take her four semesters to complete.

As we pointed out, current ontology languages do not have the representational power to describe such ontologies. These languages target simple typed ontologies where data is completely and consistently specified. In order to adequately represent and handle our example domain, we need a logical framework which is expressive enough to accommodate imprecisely specified concepts and not fully enforced relations between them. In [13], we suggested an extension to the current RuleML format to accommodate uncertain and/or inconsistent specifications, and showed how one truth maintenance logic [16] can be adapted to support such rules. Next, we discuss how this logical framework can be tailored to suit the needs of our example domain.

4. CONTEXT-DEPENDENT RULES

As stated above, our example domain contains two types on concepts: *generic* concepts, and *ontology-specific* concepts. The following data structure uniformly represents both types:

$A[i]^L$: $(T_1, \dots, T_n)(P_1, \dots, P_m)$, where:

1. **A** is a concept (propositional statement);
2. **i** is the “level” at which A was validated (derived). For example, all valid generic concepts have $i = 0$. All goal concepts are defined with $i = g$.
3. **LV** is the logical value of A. It can be: (i) T (logically true, i.e. the concept holds unconditionally in the domain); (ii) T^* (evidentially true, i.e. the concept holds with respect to the all known evidence associated with it), or (iii) U (uncertain, i.e. the validity of the concept is defined with respect to the accumulated so far partial evidence).
4. $(T_1, \dots, T_n)(P_1, \dots, P_m)$ defines the evidence associated with A as follows:
 - (i) T_1, \dots, T_n is called the *T-set* for A. It contains the required evidence that must be present to in order to consider A. In our example domain, both generic and ontology-specific concepts can be members of the T-set.
 - (ii) P_1, \dots, P_m is called the P-set for A. It defines the additional evidence for A, which if present, will further enforce its validity. In our example domain, only generic concepts can be members of the P-set.

Relations between concepts are expressed by the following two types of rules:

- Firm (monotonic) rules, or *T-rules*. These have the form $(T_1, \dots, T_n) \rightarrow A[i]^L$, and require all T-premises to match already supported concepts.

- Plausible rules, or *P-rules*. These have the form $(T_1, \dots, T_n)(P_1, \dots, P_m) \rightarrow A[i]^U$ and the validity of their conclusions is defined by the evidence associated with them.

The evidence associated with a given concept defines the *context* with respect to which that concept is evaluated. For example, consider the following rule representing relation 1A from Section 3:

$$(UG1, UG2)(UG3, UG4) \rightarrow GR1A[i]^U$$

The conclusion, GR1A, can be derived in the following three contexts:

1. All T-premises hold, but none of the P-premises holds. This defines the *minimal context*, and therefore the validity associated with the conclusion is nominal.
2. All T-premises and all P-premises hold. This defines the *maximal context*, and therefore the validity of the conclusion is the highest.
3. All T-premises and some of the P-premises hold (either the student have UG3 or UG4). T-premises, along with the satisfied P-premises, define the *context* in which the conclusion holds, and its nominal validity is further enforced by the satisfied P-premises.

To implement the notion of context-dependency, the following *duplicate rules* are created for each P-rule:

- $(R_1, \dots, R_n, P_1, \dots, P_m) \rightarrow A[i]^{T^*}$. This rule, called the *T-duplicate* of the original P-rule, captures the case where all relevant evidence for A is accumulated. It is important to note that T-duplicates are not logically equivalent to T-rules, because they may not define the complete evidence for A.
- For any $\{i_1, \dots, i_k\} \subseteq \{1, \dots, m\}$, $(T_1, \dots, T_n, P_{i_1}, \dots, P_{i_k}) \rightarrow A[i]^U$. These are called *P-duplicates* of the original P-rule, and they define all possible contexts in which A holds with different degree of truthfulness.

An ontology, O_n , is defined as a triple $\langle \{A_i\}, \{R_j\}, \{G_k \subseteq A_i\} \rangle$, where $\{A_i\}$ is a set of assumption concepts (only generic concepts can belong to this set), $\{R_j\}$ is a set of T-rules, P-rules, T- and P- duplicates, and $\{G_k\}$ is a set of goal concepts that must to be held in the transitive closure of O_n , $TC(O_n)$. The following “relaxed” version of the inference procedure described in [13] computes the latter:

Initial step. Given $\{A_i\}_0$, $\{R_j\}$, and $\{G_k \subseteq A_i\}_0$

1. Compute $\{A_i\}_1$ by augmenting $\{A_i\}_0$ with the conclusions of all applicable rules from $\{R_j\}$. Set the level of all derived conclusions to 1.
2. If none of the rules from $\{R_j\}$ is applicable, identify those with unsatisfied generic premises and enforce the corresponding generic concepts with logical value T^* , empty T- and P-sets and level of 0. Apply the first applicable rule, and mark enforced generic concepts from its T-set as “enforced premises”.

Recursive step i.

3. Check for unsatisfied goals. If all goal concepts are reached, stop. Otherwise, compute $\{A_i\}_m$ by augmenting $\{A_i\}_{m-1}$ with the conclusions of all applicable rules from $\{R_i\}$. Set the level of all derived conclusions to i .
4. If none of the rules from $\{R_i\}$ is applicable, identify those with unsatisfied generic premises and enforce the corresponding generic concepts with logical value T^* , empty T- and P-sets, and level $(i - 1)$. Apply the first applicable rule, and mark enforced generic concepts from its T-set as “enforced premises”.

Notice that $TC(O_n)$ may contain multiple entries of the same concept supported by different T- and P-sets. Taking into account the domain semantics, we can safely ignore all but the earliest derivation of a given concept. This is similar to the assumption behind hill-climbing search, which states that once a particular search state is reached, the path to that state becomes irrelevant.

Removing all redundant entries from $TC(O_n)$, the remaining ones comprise the so-called **core set** of $TC(O_n)$. The following query-relevant information can be derived from the latter:

- The **common context** with respect to which a query is evaluated. This is defined as a triple $\langle \{EP_i\}, \{RP_n\}, \{DP_m\} \rangle$, where $\{EP_i\}$ is a set of enforced premises, $\{RP_n\}$ is a set of required generic premises, and $\{DP_m\}$ is a set of desired generic premises.
- The **topological ordering** of the concepts defined by the levels of the respective formulas. The level of the last concept in the topological ordering defines the **adjusted depth** of the ontology.

Comparing the corresponding common contexts and topological orderings of competing ontologies, the described inference procedure can answer example queries as defined in Section 3 according to the following definition.

Definition. Ontology A **dominates** ontology B iff:

- a) The adjusted depth of ontology A is less than the adjusted depth of ontology B, or
- b) The adjusted depths of both ontologies are the same, but cardinalities of the respective subsets of their common contexts compare as follows:
 - a. $|EP_i|_A < |EP_i|_B$
 - b. $|RP_i|_A > |EP_i|_B$
 - c. $|DP_i|_A < |DP_i|_B$

In the next section, we illustrate how the described logical framework supports the example domain described in Section 3.

5. EXAMPLE CONTINUED

Expressing the relations between concepts (see Figures 1 and 2) into context-dependent rules is straightforward.

Ontology A:

- 1A: $(UG1, UG2)(UG3, UG4) \rightarrow GR1A[i]^U$
- 1A-P1: $(UG1, UG2, UG3)(UG4) \rightarrow GR1A[i]^U$
- 1A-T: $(UG1, UG2, UG3, UG4)() \rightarrow GR1A[i]^{T^*}$
- 2A: $(UG6, UG7)(UG8) \rightarrow GR2A[i]^U$
- 2A-T: $(UG6, UG7, UG8)() \rightarrow GR2A[i]^{T^*}$
- 3A: $(GR1A, UG8)() \rightarrow GR4A[i]^T$
- 4A: $(UG4, GR1A)(UG9) \rightarrow GR3A[i]^U$
- 4A-T: $(UG4, GR1A, UG9)() \rightarrow GR3A[i]^{T^*}$
- 5A: $(GR4A)() \rightarrow GR2A[i]^T$
- 6A: $(GR3A, UG6)() \rightarrow GR4A[i]^T$
- 7A: $(GR4A)() \rightarrow GR5A[i]^T$
- 8A: $(GR3A)(UG10) \rightarrow GR6A[i]^U$
- 8A-T: $(GR3A, UG10)() \rightarrow GR6A[i]^{T^*}$

Ontology B:

- 1B: $(UG1, UG2)(UG3) \rightarrow GR1B[i]^U$
- 1B-T: $(UG1, UG2, UG3)() \rightarrow GR1B[i]^{T^*}$
- 2B: $(UG9)(UG7) \rightarrow GR2B[i]^U$
- 2B-T: $(UG9, UG7)() \rightarrow GR2B[i]^{T^*}$
- 3B: $(GR2B, UG4)(UG6) \rightarrow GR3B[i]^U$
- 3B-T: $(GR2B, UG4, UG6)() \rightarrow GR3B[i]^{T^*}$
- 4B: $(GR1B, GR2B)() \rightarrow GR3B[i]^T$
- 5B: $(GR2B, UG9)(UG8) \rightarrow GR4B[i]^U$
- 5B-T: $(GR2B, UG9, UG8)() \rightarrow GR4B[i]^{T^*}$
- 6B: $(GR1B, UG6)() \rightarrow GR3B[i]^T$
- 7B: $(GR3B, GR4B)() \rightarrow GR5B[i]^T$
- 8B: $(GR2B, GR3B)() \rightarrow GR6B[i]^T$

Given $A_0 = \{UG1[0]^T, UG2[0]^T, UG3[0]^T, UG4[0]^T, UG6[0]^T, UG9[0]^T\}$ (Peter’s query), $TC(A)$ contains the following derived concepts:

- $$\begin{aligned} &\{GR1A[1]^{T^*}: (UG1, UG2, UG3, UG4)(), \\ &GR3A[2]^{T^*}: (GR1A, UG4, UG9)(), \\ &GR4A[3]^{T^*}: (GR3A, UG6)(), \\ &GR6A[4]^U: (GR3A)(UG10), \\ &GR2A[4]^{T^*}: (GR4A)(), \\ &GR5A[4]^{T^*}: (GR4A)() \} \end{aligned}$$

Here the common context associated with $TC(A)$ is $\{ \{ \}, \{UG1, UG2, UG3, UG4, UG6, UG9\}, \{UG10\} \}$.

Given $B_0 = \{UG1[0]^T, UG2[0]^T, UG3[0]^T, UG4[0]^T, UG6[0]^T, UG9[0]^T\}$ (Peter’s query), $TC(B)$ contain the following derived concepts:

- $$\begin{aligned} &\{GR1B[1]^{T^*}: (UG1, UG2, UG3)(), \\ &GR2B[1]^U: (UG9)(UG7), \\ &GR3B[2]^{T^*}: (GR2B, UG4, UG6)(), \\ &GR4B[2]^U: (GR2B, UG9)(UG8), \\ &GR3B[2]^{T^*}: (GR1B, GR2B)(), \\ &GR3B[2]^{T^*}: (GR1B, UG6)(), \\ &GR5B[3]^{T^*}: (GR3B, GR4B)(), \\ &GR6B[3]^{T^*}: (GR2B, GR3B)() \} \end{aligned}$$

Here the common context is the following:

- $$\{ \{ \}, \{UG1, UG2, UG3, UG4\}, \{UG7, UG8\} \}$$

To decide which program is better for Peter, compare the corresponding common contexts and topological orderings of concepts from the core sets of ontologies A and B. Notice that the adjusted depth of ontology B is less than the adjusted depth of ontology A. This suggests that if Peter is primarily interested in completing the core courses in the shortest amount of time, program B is the better choice for him. However, if Peter is more concerned with finding the best match for his undergraduate background, program A will be a better choice because $|RP_{iA}| > |RP_{iB}|$ and $|DP_{iA}| < |RP_{iB}|$.

In Ana's case, given $A_0 = \{UG1[0]^T, UG2[0]^T, UG3[0]^T, UG6[0]^T, UG7[0]^T, UG8[0]^T\}$, $TC(A)$ contains:

```
{ UG4[0]^T: ()(),
GR1A[1]^U: (UG1, UG2, UG3) (UG4),
GR2A[1]^T*: (UG6, UG7, UG8) (),
GR4A[2]^T*: (GR1A, UG8) (),
GR3A[2]^U: (GR1A, UG4) (UG9),
GR2A[3]^T*: (GR4A) (),
GR5A[3]^T*: (GR4A) (),
GR6A[3]^U: (GR3A) (UG10) }
```

Here the common context is the following:

```
{ {UR4}, {UG1, UG2, UG3, UG6, UG7, UG8}, {} }.
```

Similarly, given $B_0 = \{UG1[0]^T, UG2[0]^T, UG3[0]^T, UG6[0]^T, UG7[0]^T, UG8[0]^T\}$, $TC(B)$ contains:

```
{GR1B[1]^T: (UG1, UG2, UG3) (),
UG9[1]^T*: ()(),
GR3B[2]^T*: (GR1B, UG6) (),
GR2B[2]^T*: (UG9, UG7) (),
GR3B[3]^T*: (GR1B, GR2B) (),
GR4B[3]^T*: (GR2B, UG9, UG8) (),
GR6B[3]^T*: (GR3B, GR4B) (),
GR5B[4]^T*: (GR3B, GR4B) () }
```

Here the common context is:

```
{ {UG9}, {UG1, UG2, UG3, UG6, UG7, UG8}, {} }.
```

Clearly, program A will be the better choice for Ana, because the respective ontology has the lesser adjusted depth, and all subsets of the respective common contexts have the same cardinality.

6. CONCLUSION

In this article, we have presented a logical framework utilizing context-dependent rules intended to support Semantic Web ontologies that are not fully and/or precisely specified. A hypothetical application scenario was described to illustrate the type of ontologies targeted. We advocated that to support such ontologies, the inference procedure should be able to maintain incomplete, uncertain, and even inconsistent specifications, preserving at the same time the validity and meaningfulness of derived knowledge. Because "meaningfulness" in this case can only be evaluated with respect to a particular context, we have shown how the presented logical framework identifies, maintains and interprets that context to answer specific types of queries.

Acknowledgement. This work was partially supported by a CSU-AAUP research grant.

7. REFERENCES

- [1] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. **Scientific American** 284, May, 2001, pages 34-43.
- [2] <http://www.w3.org/>
- [3] T. Bray, Paoli, J., Sperberg-McQueen, C.M., Maler, E. (editors) Extensible Markup Language (XML) 1.0, W3C Recommendation, October 2000.
- [4] O. Lassila, Swick, R. Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation, February 1999.
- [5] D. Brickley, Guha, R.V. Resource Description Framework (RDF) Schema Specification 1.0, W3C Candidate Recommendation, March 2000.
- [6] F. van Harmelen, Patel-Scheider, P. F., Horrocks, I. (editors) Reference description of the DAML + OIL Ontology markup language (contributors: T. Berners-Lee, D. Brickley, D. Connolly, M. Dean, S. Decker, P. Hayes, J. Heflin, J. Hendler, O. Lassila, D. McGuinness, L.A. Stein), March 2001.
- [7] M. Dean, Connolly D., van Harmelen, F., Hendler, J., Horrocks, I., D. McGuinness, Patel-Scheider, P.F., L.A. Stein (editors) OWL Web Ontology Language 1.0 Reference, 2002.
- [8] <http://www.ruleml.org/>
- [9] C. Damasio, Pan, J., Stoilos, G., and Straccia, U. Representing Uncertainty in RuleML. **Fundamenta Informaticae**, pages 1 – 24, IOS Press, 2007.
- [10] T. Lukasiewicz, Expressive Probabilistic Description Logics. **Artificial Intelligence**, 172, (6-7), pages 852 – 888, 2008.
- [11] Z. Huang, van Harmelen, F., and ten Teije, A. Reasoning with Inconsistent Ontologies: Framework, Prototype, and Experiment. **Semantic Web Technologies: Trends and Research in Ontology-Based Systems** (eds. John Davies, Rudi Studer, Paul Warren), John Wiley & Sons, 2006.
- [12] S. Schobach and Corner, R. Non-standard Reasoning Services for the Debugging of Description Logic Terminologies. In **Proc. International Joint Conferences on Artificial Intelligence'2003**, 2003.
- [13] N. Zlatareva, Supporting Uncertainty and Inconsistency in Semantic Web Applications. In **Proc. 21st International FLAIRS Conference**, AAAI Press, 2009.
- [14] G. Antoniou and van Harmelen, F. **A Semantic Web Primer**, MIT Press, 2007.
- [15] J. Doyle, A Truth Maintenance System. **Artificial Intelligence**, 12, pages 231 – 272, 1979.
- [16] Popchev I., Zlatareva, N., and Mircheva, M., A Truth Maintenance Theory: An Alternative Approach. In **Proc. 9th European Conference on Artificial Intelligence (ECAI'90)**, 1990.