Local Closed-World Assumptions for reasoning about Semantic Web data*

Elisa Bertino[1], Alessandro Provetti[2] and Franco Salvetti[3]

[1] Dipartimento d'Informatica e Comunicazione
Università degli studi di Milano. Milan, I-20135 Italy
`bertino@dico.unimi.it`.
[2] Dipartimento di Fisica
Università degli studi di Messina. Messina, I-98166 Italy
`ale@unime.it`
[3] Dept. of Computer Science
University of Colorado at Boulder. Boulder, CO 80309-0430 U.S.A.
`franco.salvetti@colorado.edu`

Abstract The Semantic Web (SW) can be seen as abstract representation and exchange of data and metadata. Metadata is given in terms of data mark-up and reference to shared, Web-accessible ontologies. Several interesting languages are now available for the Semantic Web. They exploit XML allowing data/metadata communication, yet are endowed with a logical semantics. Such languages allow compact descriptions by means of inheritance mechanisms that permit one to describe an object as belonging to one or more classes whose hierarchy is already described on the Web. With few exceptions, SW logical languages are designed to be monotonic, thus they cannot employ the closed-world assumption i) to make object description compact and most importantly ii) to prevent large ontologies from admitting inconsistency resulting from multiple inheritance. We address these problems by proposing a version of Local Closed-World Assumption that fits SW purposes. Its extent is itself the subject of negotiation between communicating agents. In this light, we give a new default semantics to RDF type inheritance primitives and show that Answer Set Programming seems a promising tool for the automation of consistency maintenance over Semantic Web annotations.

1 Introduction

This article is about applying Answer Set Programming (ASP) to certain aspects of the Semantic Web, namely agent communication.

Answer Set Programming, in brief, is a confluence of Deductive Databases and Logic Programming. ASP programs have DATALOG syntax with extension

* This work has been supported by i) MIUR COFIN project *Aggregate- and number-reasoning for computing: from decision algorithms to constraint programming with multisets, sets, and maps.* and ii) the Information Society Technologies programme of the European Commission, Future and Emerging Technologies under the IST-2001-37004 WASP and IST-2001-33058 PANDA projects.

to disjunction and default negation, and Gelfond-Lifschitz declarative semantics [Gelfond and Lifschitz, 1991]. ASP allows declarative problem-solving based on the application of *default* rules, i.e., the drawing of conclusions based on lack of evidence of the contrary, thus capturing the notion of *typical conclusion*. Thanks to defaults, ASP is a suitable language for expressing complicated or under-defined problems in a very concise form. Nowadays, there are rather efficient solvers [Systems] that can compute the answer sets of programs defining thousands of atoms within few seconds. The formal description of ASP can be found in the original works of Gelfond and Lifschitz [Gelfond and Lifschitz, 1991] and in the literature accompayining the ASP solvers [Systems].

The Semantic Web has been defined (Hendler) as

> [...] the abstract representation of data on the World Wide Web, based on the RDF standards and other standards to be defined.

and [Berners-Lee et al., 2001] as

> [..] an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.

Developed by the W3C, in collaboration with a large number of researchers and industrial partners, we believe that the Semantic Web is a place where results from Deductive databases, Knowledge Representation and Nonmonotonic Reasoning Techniques can be applied successfully, and their effectiveness carefully evaluated.

Of course, we are neither alone nor the originators of this scientific stance. The main result of formal AI techniques applied to the Semantic Web is the Description Logics (DL) semantics given to the SW representation languages RDF-Schema Language (and following layers such as DAML+OIL built on top of it).

Our work, while acknowledging the firm results obtained by the Description Logic community, describes a somewhat parallel approach, whose aim is twofold. First, by discussing the languages used in the SW, notably DAML+OIL, in a nonmonotonic reasoning framework, we rephrase them in a more concise, easy-to-grasp form likely to bring them to the attention of people with a background in Logic Programming. Second, we argue that, like any large ontology based on inheritance, the Semantic Web allows contradictory conclusions to be drawn, unless the chosen semantics accounts for defaults and nonmonotonism.

We propose a solution for this problem that is based on the translation of DAML+OIL statements into an ASP program and its execution. Similar proposals have been recently put forward, e.g. by Grosof [Grosof, 2002]. In this work we pursue an original approach with the goal of comprehending and reconstructing the Semantic Web in terms of Answer Set Programming.

1.1 The case for non-monotonic reasoning in the SW

So far, the knowledge-representation aspect of languages for the SW has been addressed by the development of monotonic formalisms. There are good reasons for this choice, which seems to reflect the views of Tim Berners-Lee and other W3C contributors. A description of the same object, found elsewhere on the Web, should not alter the properties we are assigning to *our* object. Indeed, locality of the reasoning is important. Default assumptions, unfortunately, depend on the global state of the representation, so they cannot in general ensure that our conclusions are not overridden. On the Web site of the World Wide Web Consortium, one can access documents defining the *official* declarative semantics of RDF and RDF(S) (schema), two languages which form the basis of DAML+OIL, which is discussed next. The declarative semantics of RDF appears to be monotonic, and one could evince from the following quotation[1]:

> RDF is an assertional logic, in which each triple expresses a simple proposition. This imposes a fairly strict monotonic discipline on the language, so that it cannot express closed-world assumptions, local default preferences, and several other commonly used non-monotonic constructs.

However, it seems to us that when object description is made against vast, heterogeneous *global ontologies,* default inheritance is exactly the type of inheritance we need. Moreover, the closed-world assumption, which would be senseless and impossible to compute against the full Web, can be triggered in a way that reconciles default inheritance without bringing in nonmonotonic effects that may be undesirable.

2 Introduction to SW representation languages

According to Hendler:

> The Semantic Web is based on machine-readable descriptions of information, linked to ontologies which define the terms.

In other words, terms may be communicated unambiguously by means of metadata (mark-up) that links the term to a set of universally-agreed, Web-accessible definitions that, combined, provide the *intended meaning* of the term. We do not need to discuss here how elusive this desirable goal is. Rather, we move on to outline the state of the art in SW languages, i.e., languages, broadly speaking, proceeding from XML whose goal is to support exchange of machine-readable data.

[1] The text presented here is from the W3C document: *RDF Semantics, W3C Working Draft 23 January 2003* available at *http://www.w3.org/TR/2003/WD-rdf-mt-20030123/.*

3 An Answer Set-style semantics for DAML+OIL

Nowadays the accepted semantics for DAML+OIL is a strictly non-monotonic one, even if it is in a way clear that the Web is not itself a monotonic object. At the same time it is easy to understand that the compactness of using defaults rules cannot be exploited by the current idea of Semantic Web. This inability leads to an enlargement of the size of the pages. The two strictly and related concepts of non-monotonic and default reasoning are in a way solved by introducing a new explicit semantics for `daml:type` and `daml:subClassOf`. Please notice that `type,` strictly speaking, is defined in the RDF, i.e., the language underlying DAML+OIL. However, the RDF definition is imported inside DAML+OIL with the following definition, found in the official DAML+OIL Web page *http://www.daml.org/2001/03/daml+oil*

```
<rdf:Property rdf:ID="type">
   <samePropertyAs rdf:resource="http://www.w3.org/
    1999/02/22-rdf-syntax-ns\#type"/>
</rdf:Property>
```

The usage of default reasoning is subordinated by the CWA and, as we have stated, it is not reasonable to use such assumption for the web, considering that any given set of RDF statements is only part of the Web, and not the whole. Even if that is true, there are many practical situations in which a Local Closed Word Assumption (LCWA) combined with default reasoning is more than acceptable.

The domain over which the CWA is drawn, however, should be negotiated as part of SW access and use. In such situations it is up to the "agents" (programs consulting a SW marked-up page) to decide what their world is. It is likely that in a situation like the one in which one agent is a seller and the other the customer, their world will be just the union of their respective Knowledge Bases (KB) and, if necessary, some well known ontologies describing some default rules. In this case, by considering that anything could be a *resource* described into the Semantic Web, and that about any resource it is possible to make assertions, it is straightforward to imagine a set of RDF assertions stated by the two agents in which they define the set of semantic, marked-up pages relevant to their transaction (which will be their closed world).

Having such closed world it is clear that the agents will be able to exploit the idea of default reasoning for making inferences. Therefore it will be possible to find conclusions based on a set of assertions just by using defaults rules provided by an ontology or even by the agents themselves. This allows the agents to reason even without complete information, but at the same time to drop a conclusion inferred by default whenever new, relevant knowledge is added. Moreover the description of their knowledge is compact.

This valuable schema of reasoning for the Semantic Web can be achieved *by giving to some property of DAML+OIL an interpretation in terms of defaults.* Considering that the semantics is given by a logical program written in SMODELS, we will introduce the semantics by the well known example of Pingu and its ability to fly given by default inheritance.

We intend to give ASP semantics to a set of DAML+OIL sentences by means of translation into a program made up of two modules. The first module, π_1, is generic, i.e., it has as answer sets the intended semantics of basic DAML+OIL constructor relations.

The second module,π_2, is obtained as a direct, almost one-by-one translation of RDF assertions into logic facts. At the same time the answer sets of program $\pi_1 \cup \pi_2$ will be the logical equivalent RDF statements inferred from the given KB, or, if that is the case, a set of statements representing errors in the assertions given.

In this framework the program π_1 is the part in which we can define how the meaning of a property is to be considered. By changing it, we will change the intended meaning of a property.

3.1 Default Inheritance

Consider the case in which there is the `daml:Class` of penguins that is a `daml:subClassOf` of the `daml:Class` of birds that is a `daml:sublClassOf` of the `daml:Class` of flying things. There is also the `daml:Class` of things that do not fly, which is defined as `daml:complementOf` the class of the things that fly and of course Pingu, a penguin.

```
<daml:Class rdf:ID="Bird">
  <daml:subClassOf rdf:resource="rdfs:Resource"/>
  <daml:subClassOf rdf:resource="#Flying"/>
</daml:Class>
  <daml:Class rdf:ID="Penguin">
  <daml:subClassOf rdf:resource="#Bird"/>
</daml:Class>
  <daml:Class rdf:ID="Flying">
  <daml:daml:complementOf rdf:resource="#n_Flying"/>
  <daml:subClassOf rdf:resource="rdfs:Resource"/>
</daml:Class>
<daml:Class rdf:ID="n_Flying">
  <daml:complementOf rdf:resource="#Flying"/>
  <daml:subClassOf rdf:resource="rdfs:Resource"/>
</daml:Class>
<Penguin rdf:ID="pingu">
  <daml:type rdf:resource="#n_Flying"/>
</Penguin>
```

It is well-known in SW literature that such rules could be translated in a sequence of logical facts like the following:

```
t("Bird", "daml:subClassOf", "Flying").
t("Penguin", "daml:subClassOf", "Bird").
t("pingu", "daml:type", "Penguin").
t("pingu", "daml:type", "n_Flying").
t("n_Flying", "daml:complementOf", "Flying").
t("Flying", "daml:complementOf", "n_Flying").
```

Pingu is of `daml:type` penguin, therefore exploiting the monotonic semantics of the property `daml:subClassOf` Pingu is an instance of the class of things that fly. Since that is, however, not true, we have made an RDF assertion which says explicitly that Pingu is an instance of the class of things that do not fly. Due to the monotonic semantics of `daml:complementOf`, there will be a statement into the ASP encodings of that KB saying that Pingu cannot be an instance of two classes, one disjoint from the other. To avoid this problem we can introduce a new explicit non-monotonic semantics of `daml:type` and `daml:subClassof` using an ASP program (in this case, with SMODELS syntax) as follows:

```
d(X) :- t(X,Y,Z).
d(Y) :- t(X,Y,Z).
d(Z) :- t(X,Y,Z).

triple(X,Y,Z) :- t(X,Y,Z).

subClassOf(X,Y) :-
   d(X),
   d(Y),
   triple(X, "daml:subClassOf", Y).

type(X,Y) :-
   d(X),
   d(Y),
   triple(X, "daml:type", Y).

triple(S, "daml:subClassOf", O) :-
  d(S),
  d(O),
  d(B),
  d(C),
  triple(S, "daml:subClassOf", B),
  triple(B, "daml:subClassOf", O),
  not cannotBeSubClassOf(S,O).

cannotBeSubClassOf(X,C) :-
  d(X),
  d(C),
  d(A),
```

```
  triple(X, "daml:subClassOf", A),
  triple(A, "daml:complementOf", C).

triple(S, "daml:type", O) :-
  d(S),
  d(C),
  d(B),
  d(O),
  triple(S, "daml:type", B),
  triple(B, "daml:subClassOf", O),
  not cannotBeTypeOf(S,O).

cannotBeTypeOf(X,C) :-
  d(X),
  d(C),
  d(A),
  triple(X, "daml:type", A),
  triple(A, "daml:complementOf", C).
```

The ASP program above is ready to be fed to an ASP solver. For instance, if we run SMODELS on it, the result will be as follows:

```
smodels version 2.26. Reading...done
Answer: 1
Stable Model:
type("pingu","n_Flying")
type("pingu","Penguin")
type("pingu","Bird")
subClassOf("Penguin","Flying")
subClassOf("Bird","Flying")
subClassOf("Penguin","Bird")
```

The possible existence of Magic, a penguin which flies, can be captured by the new non-monotonic semantics because there is a direct way to infer that it flies, either by an explicit directed arch that says that it flies or by inference through the semantics of `daml:subClassOf`. In this semantics, by default, any other instance of classes that are subclasses of the class of the birds will be considered an object that flies, avoiding the necessity of writing any statement relative to the object's ability to fly. This new semantics not only gives a simple method of dealing with default reasoning, but also a way to treat the inevitable existence of exceptions in any system of classification. The idea behind this semantic is that any conclusion thrown by default has to be dropped if an explicit opposite knowledge is found.

4 Local Closed-World Assumption Assumption

The Closed World Assumption (CWA) can be described as a rule of thumb (extra-logical) by which what cannot be proved true is assumed to be false. This assumption is based on the idea that the prover is omniscient, so if he/she/it cannot prove ϕ then ϕ has to be false. For lack of space, we cannot discuss the CWA here; the reader is invited to consult the the non-monotonic reasoning and deductive database literature. Here we would like to introduce the following perspective. Let the prover note that his/her/its own knowledge is de-facto limited *but relevant to* ϕ. In the SW the limitation is the *Web horizon.* Two, non-exclusive types of horizons are considered:

- trust: only reliable/verified, or even internal resources (pages) are considered as part of the theory against which deduction is performed. The obvious practical utility is to avoid inconsistency as the result of malicious/unchecked behavior.
- reachability: the collection of all resources that can be reached within a given maximal amount of time.

Therefore, we propose the Local Closed-World Assumption (LCWA) as the set of all resources that are considered part of the theory. The LCWA must be declared about resources with an appropriate syntax. In its extreme form, the LCWA could correspond to limiting deduction to the sole document at hand.

5 Relation to literature

One approach that seems similar to our is that of [Heflin & Muñoz-Avila, 2002], who introduce the LCWA in order to apply planning as the deduction mechanism underlying Semantic Web services. Another similar approach is being pursued by Grosof [Grosof, 2002]. At the moment we are not able to carefully assess the relative differences between these approaches and ours. It seems however that our perspective is narrower, i.e., we focus on a non-standard meaning for `daml:subClassOf`, which potentially makes a greater difference with the standard DL logic semantics for RDF.

6 Open issues

One aspect of non-monotonic reasoning for the Semantic Web that has not been addressed here is scalability. There is no doubt that scalability, in the end, will be a crucial factor in the future developments of the SW. In this sense, Description Logics are advantageous because their deductive complexity is well-studied and can be somewhat controlled.

So far, there is no literature about any application of non-monotonic systems, such as ASP solvers, to such a vast theory as the Semantic Web. On the other hand, we believe that the vastness of the Web today is also an argument in favor of our approach, since vastness means that the general case is a de-facto incomplete-information case, and no reasoner, human or otherwise, would restrict him/her/itself to conclusions based on evidence alone. Defaults, therefore, seem more to clarify the picture than to make it more complex.

Acknowledgments

This work was motivated by discussions with Sheila McIlraith and Tran Cao Son. Thanks also to Stefania Costantini, Michael Gelfond and the AI group in Corunna for useful discussions on these topics.

References

[Berners-Lee et al., 2001] Berners-Lee T., Hendler J. and Lassila O., 2001. *The Semantic Web.* Scientific American, US edition, May 2001 issue.

[Cadoli et al., 1997] Cadoli M., Eiter T. and Gottlob G.,, 1997. *Default Logic as a Query Language.* IEEE Transactions on Knowledge and Data Engineering.

[Cholewiǹski et al., 1999] Cholewiński, P, Marek, W., Mikitiuk A., and Truszczyński, M., 1999. *Computing with default logic,* Artificial Intelligence Journal, 112, pp. 105–146.

[Eiter et al., 1997] Eiter T., Gottlob G. and Mannila H., 1997. *Disjunctive Datalog.* ACM Transactions on Databases, 22(3):364–417.

[Heflin & Muñoz-Avila, 2002] Heflin, J. and Muñoz-Avila, H. *LCW-Based Agent Planning for the Semantic Web.* AAAI02 Workshop on Ontologies and the Semantic Web. AAAI Press, pp. 63–70.

[Garey & Johnson, 1979] Garey M. and Johnson D., 1979. *Computers and Intractability. A Guide to the Theory of NP-Completeness.* Freeman and Co.

[Gelfond and Lifschitz, 1991] Gelfond, M. and Lifschitz, V., 1991. *Classical negation in logic programs and disjunctive databases.* New Generation Computing: 365–387.

[Grosof, 2002] Grosof, B, 2002. *Rules + Ontologies for Semantic Web Services.* U. Maryland seminar talk. Available from the Web at *http://ebusiness.mit.edu/bgrosof/#PapersEtc*

[Hawke, 2002] Hawke, S., 2002 *Semantic Web Languages: Fundamentals and Combinations* W3C Tech. report. *http://www.w3.org/2002/08/LX/v3/paper*

[Marek & Truszczyński, 1999] Marek, W., and Truszczyński M., 1999. *Stable models and an alternative logic programming paradigm.* The Journal of Logic Programming.

[McIlraith et al., 2001] McIlraith, S., Tran Cao S. and Zeng, H., 2001. *Semantic Web Services.* IEEE Intelligent Systems, Special Issue on the Semantic Web. IEEE Press.

[McIlraith & Tran Cao, 2002] McIlraith, S. and Tran Cao S., 2002. *Adapting Golog for Composition of Semantic Web Services.* Proc. of KR'2002 Conference. Morgan Kauffman.

[RuleMl] The RuleML initiative *http://www.dfki.uni-kl.de/ruleml/*

[Schlipf, 1995] Schlipf, J., 1995. *The expressive powers of the logic programming semantics.* Journal of Computer and Systems Sciences 51, pp. 64–86.

[Swift, 2002] Swift, T., 2002. *OWL + ASP = DeadColdFish.* Dagstuhl Seminar.

[Systems] Web location of the most known ASP solvers.
Cmodels: *http://www.cs.utexas.edu/users/yuliya/*

aspps: *http://www.cs.uky.edu/ai/aspps/*
DLV: *http://www.dbai.tuwien.ac.at/proj/dlv/*
Smodels: *http://www.tcs.hut.fi/Software/smodels/*