Reasoning about RDF statements with default rules

submitted to W3C Workshop on Rule Languages for Interoperability *

Elisa Bertino Alessandro Provetti Franco Salvetti

CS Dept. & CERIAS - Purdue University. West Lafayette, IN 47907-2086

bertino@cerias.purdue.edu.

Physics Dept. - Università di Messina. Messina, I-98166 Italy

ale@unime.it

CS Dept. - University of Colorado at Boulder. Boulder, CO 80309-0430

franco.salvetti@colorado.edu

Abstract

We propose a version of Local Closed-World Assumption that is well-suited for logic-based data and knowledge integration from disparate Web sources and for agent communication over the Web. To do so, we propose a default-based interpretation of the basic type inheritance primitives of RDF in terms of Answer Set Programs. Translating such statements into Answer Set Programming gives them an alternative declarative semantics and makes it possible to apply the inferential engines now available for Answer Set Programming. Our proposal is orthogonal to the current development of Semantic Web languages but it can help by i) making object description compact and most importantly ii) preventing large data sets from inconsistencies arising from multiple inheritance.

1 Introduction

This position paper concern applying default rules to automated reasoning about RDF data. Non-monotonic reasoning is based on Closed-world assumption. Closed-world assumption against the whole Web is clearly unviable, yet it can hardly be done without when we consider agents negotiating some communication agreement.

With our experiments [Bertino et al., 2003] in embodying localized closed-world assumptions in agent communicating 'in' RDF we have shown that the language of Answer Set Programming (a form of logic programming) is viable for such use. One key factor is that Gelfond-Lifschitz semantics of ASP statements, aptly called rules, sees them axiomatically, i.e., as constraints on a set of beliefs a rational agent can holds, and not as truth-functional formulas. Therefore, we outline a research program intended to work out the theoretical framework in which ASP-based localized closed-world assumption can be employed in communication and negotiation in the Semantic Web. We propose a version of Local Closed-World Assumption that is well-suited for logic-based data and knowledge integration from disparate sources and for agent communication. To do so, we propose a default-based interpretation to certain RDF type inheritance primitives in Answer Set Programming. The translation into Answer Set Programming (ASP) gives a declarative semantics to a relevant fragment of RDF and makes it possible to apply the inferential engines now available for ASP, e.g. the DLV engine [Eiter et al., 1997]. Our proposal is orthogonal to the current development of Semantic Web languages but it can help by i) making object description compact and most importantly ii) preventing large data sets from inconsistency resulting from multiple inheritance.

Answer Set Programming, in brief, is a confluence of Deductive Databases and Logic Programming. ASP programs have DATALOG syntax with extension to disjunction and default negation, and Gelfond-Lifschitz declarative semantics [Gelfond and Lifschitz, 1991]. ASP allows declarative problem-solving by 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 accompanying the ASP solvers [Systems].

The research presented here has been developed with the aim of supporting the development of *Web applications* based on computational logic. In this respect, we understand *data integration* as integration of Web sources while agent communication as Web agents negotiating the terms of their own communication. Indeed, the underlying representation language used here in RDF, a version of XML which is intended for the Web. However, we stress that our work remains orthogonal to the current development of Semantic Web languages. The Semantic Web has been developed by the W3C, in collaboration with a large number of researchers and industrial partners. Hendler defines it 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

*This work has been supported by the Information Society Technologies programme of the European Commission, Future and Emerging Technologies under the IST-2001-37004 WASP project.

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

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.

The main result of logic-based AI techniques applied to the development of the Semantic Web is perhaps the Description Logics (DL) semantics given to the SW representation languages RDF-Schema and to *layers* (such as [RDF, 2003], DAML+OIL and the more recent [OWL, 2003]) built on top of it.

Our work, while acknowledging the firm results obtained by the Description Logic community and the W3C, 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. Indeed, only very recently has this subject been considered in the literature [Eiter et al., 2004b; Eiter et al., 2004; Eiter et al., 2004b]. In particular, [Eiter et al., 2004] is the first systematic attempt at bridging the W3C semantics, which is based on Description logic and traditional minimal model semantics of Logic programming and Nonmonotonic reasoning.

From the beginning of their development, SW languages have been given a strictly monotonic interpretation, i.e, once asserted a statement is never retracted. 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 An Answer Set-based interpretation for DAML+OIL statements

Nowadays the accepted semantics for DAML+OIL language 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 [Eiter et al., 2004]. 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[2]

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

Since the application of default rules requires making an assumption that *the world is closed*, i.e., that it can be completely inspected, such mechanism is clearly not applicable for reasoning about Web data *as a whole*. What can be done, however, is the application of a Closed-world assumption local to an arbitrary but well-defined (distributed, yet accessible) data set

We foresee several practical situations where a Local Closed Word Assumption (LCWA) combined with default reasoning brings in an acceptable computational burden.

The domain over which the LCWA 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 marked-up page, not necessarily a Semantic Web one) to decide what their *world* is. It is likely that in a one-to-one session such as in e-commerce (one agent is a seller and another the customer), the world will be just the union of their respective Knowledge Bases (KB) plus, when necessary, some well-known ontologies describing the default rules. Readers with familiarity with the Semantic Web can imagine, given anything could be a *resource* described into the SW, and given

[1] This quote 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/*.

[2] *http://www.daml.org/2001/03/daml+oil*.

that for almost any such resource it is possible to make assertions, 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. Such set of assertions will be their *world*.

In small, closed worlds, agents will be able to exploit the idea of default reasoning for making inferences. And to handle a *compact* sets of statements.

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 with incomplete information. However, agents must be ready to drop a conclusion inferred by default whenever new, relevant knowledge is added. 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* attached to a property. By changing it, we will change the intended meaning of a property.

2.1 Default Inheritance

We show a simple example on the implementation of default inheritance. Even though the example itself can be seen in undergraduate AI textbooks by now, we believe that the solution presented here is of large, if not general, applicability.

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 disjoint classes. To avoid this problem we can introduce a new explicit non-monotonic semantics of `daml:type` and `daml:subClassof` using the following ASP rules[3]:

```
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),
```

[3] In the following description the syntax of the SMODELS inferential engine will be used. However, only few changes are needed to make the program compatible with a different engine.

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

The ASP program consisting of the basic statements and of the rules above can be fed to an ASP solver. For instance, if we run the SMODELS solver on the 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 that 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 reached by default won't hold anymore when the Web agent finds new, explicit opposite knowledge.

3 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 LCWA here; the reader is invited to consult the the non-monotonic reasoning and deductive database literature, as for instance in Eiter et al. work (2003, 2004, 2004b).

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.

4 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]. 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.

5 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 only very recent literature about any application of non-monotonic systems, such as ASP solvers, to such a vast theory as the Semantic Web, with the exception of Eiter et al. (2003, 2004a, 2004b). One approach close in spirit to ours is that of [Farrugia, 2003] who seems to advocates the search for a model-theoretic semantics of RDF statements that reflects the human interpretation of each statement as close as possible. 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.

Another topic which needs to be addressed is the relationship with the SPARQL[4]query language for RDF. We believe that the solution for applying default reasoning (so called intensional rules) to RDF statements can be greatly simplyfied if we restrict to accessing RDF data throughtout the SPARQL query language. This way, the (more or less) standard schema of non-monotonic reasoning on top of *monotonic* data can be adopted in our framework:

- RDF relations defined by posted data are considered extensional, in short they can be used to do reasoning but their extension remains fixed.
- fresh relations, defined in the context of local communication are deemed intensional predicates and are computed by applying ASP rules.

Acknowledgments

Thanks to Sheila McIlraith, Tran Cao Son, Stefania Costantini and Michael Gelfond for useful discussions on these topics.

[4]http://www.w3.org/TR/2005/WD-rdf-sparql-query-20050217/

References

[Berners-Lee et al., 2001] Berners-Lee T., Hendler J. and Lassila O., 2001. *The Semantic Web.* Scientific American, 184(5), pp. 34–43.

[Bertino et al., 2003] Bertino E., Salvetti F. and Provetti A., 2003. *Local Closed-world Assumptions for reasoning about Semantic Web data.* Proc. of APPIA-GULP-PRODE joint conference on declarative programming, pp. 314–323.

[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.

[Eiter et al., 2004] Eiter T., Lukasiewicz T., Schindlauer R. and Tompits H., 2004. *Combining Answer Set Programming with Description Logics for the Semantic Web.* Proc. of KR 2004. Morgan Kauffman, pp. 141–151.

[Eiter et al., 2004b] Eiter T., Lukasiewicz T., Schindlauer R. and Tompits H., 2004. *Well-Founded Semantics for Description Logic Programs in the Semantic Web.* Proc. of RuleML 2004, pp. 81–97.

[Eiter et al., 2004b] Eiter T., Fink M., Greco G. and Lembo D., 2003. *Efficient Evaluation of Logic Programs for Querying Data Integration Systems.* Proc. of ICLP 2003 Conference. Springer LNCS, pp. 163–177

[Farrugia, 2003] Farrugia J., 2003. *Model-Theoretic Semantics for the Web.* Proc. of WWW2003, 12th Int'l Conference on World Wide Web. ACM Press, pp. 29–38.

[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.

[OWL, 2003] Patel-Schneider P., Hayes, P. and Horrocks, I., editors, 2003. *Web Ontology Language (OWL), Abstract Syntax and Semantics.* W3C Working Draft 3. *http://www.w3.org/TR/owl-semantics/*

[RDF, 2003] Hayes P., editor, 2003. *RDF Semantics.* W3C Working Draft. *http://www.w3.org/TR/rdf-mt/*

[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/*