ANDRZEJ SZAŁAS

SEMANTIC WEB IN A NUTSHELL[1]

STRESZCZENIE

Artykuł zarysowuje wybrane zagadnienia związane z Semantycznym Internetem. Wychodząc od problemów rozwiązań stosowanych we współczesnych wyszukiwarkach, traktuje o architekturze i zasadniczych koncepcjach Semantycznego Internetu, którego kluczowym pojęciem są ontologie. Tekst omawia miejsce logik w specyfikacji i weryfikacji ontologii, a także wnioskowaniu o nich, koncentrując się przede wszystkim na językach regułowych, w tym na języku 4QL, dostarczającym prostych konstrukcji umożliwiających reakcję na niepełną i/lub sprzeczną informację.

ABSTRACT

The paper outlines selected topics related to Semantic Web. We start with problems with the use of current web search engines. Next, we discuss the architecture and basic concepts of Semantic Web whose central ideas focus around ontologies. We show the role of logics in the specification and verification of ontologies as well as reasoning about them. We mainly concentrate on rule languages. In particular we discuss the 4QL query language supplying the user with simple, yet powerful constructs for filling gaps in missing knowledge as well as for disambiguation of inconsistencies.

TODAY'S WEB

Today's web is a huge repository of distributed and heterogeneous information sources, including web pages, contents generated from databases, etc. Typically, these sources, are not structured. The way information is presented makes it easily accessible for humans, but remains hard to understand and manipulate by computers. Web information seeking is dominated by keywords-based search engines. The meaning attached by these engines to user-supplied sequences of keywords is based on more or less complex heuristics and AI-based techniques. No matter how intelligent algorithms used for that purpose are, they can only approximate user's requirements. For example, when looking for a hotel in the centre of a given town, one can use keywords „hotel", town name, maybe a district or a region name, what usually results in hundreds of thousands, if not millions, of irrelevant pages. Keywords cannot provide rigorous semantics understood in the same way by both search engines and users. Additionally, keywords are not well related to each other.

Importantly, keyword-based search is seriously limited in its expressive power. For example, it is impossible to express a query intended to return *only* web pages on museums located in a given town. When one attempts to supply such queries, search engines usually return also (or mainly) web pages on museums in other towns as well as objects located in the town, but other than museums. Similarly, planning a trip around a given region or country, involving hotels, transport, places of interest, requires a considerable effort, especially when one wants to express some quality demands concerning hotels, airplanes, trains, etc. Keyword-based search allows one to use concepts (expressed by keywords or phrases) and set-theoretical operations on them (intersection, union, complement) or, in other words, their counterparts in propositional logic (conjunction, disjunction, negation). However, search engines frequently violate the semantics of operations on sets/propositional connectives. The reasons of such situations remain unclear. What is more serious, it is unclear what the actual underlying semantics is. This makes processing of information supplied by search engines difficult, especially in automated reasoning tasks.

When many heterogeneous subjects are involved, one additionally faces problems of low quality of information caused by the lack of knowledge, inconsistencies, uncertainty and noise.

[1] This article has been prepared while its author has been a visiting professor in Warsaw School of Computer Science.

This, of course, makes automated web-based information processing much harder.

SEMANTIC WEB

Semantic Web has been proposed by Tim Berners Lee[2] as a remedy to many drawbacks of modern search engines. The key idea of the semantic approach to web search is based on the observation that people communicate using concepts, concept hierarchies and relationships among concepts. Concept hierarchies are called *taxonomies, while* taxonomies enriched with relationships constitute *ontologies*. Therefore, a better approach to information seeking can be based on the requirement that information included in web pages should be classified by their designers according to chosen, preferably standard, ontologies.

For example, one can consider an ontology describing wines and wine producers:

- concepts: wine, red wine, white wine, desert wine, winery, region
- roles: is produced by, is located in.

A typical taxonomy of wine world can specify that red wine, white wine and desert wine are subconcepts of the concept 'wine'. They can also specify particular wines, wineries and regions:

- wines: Trifecta, Beaucanon Estate Chardonnay
- wineries: Andretti Winery, Beaucanon Estate Winery, Chateau Brethous
- regions: Bordeaux, California,

together with relationships among them, like:

- Trifecta is produced by Beaucanon Estate Winery
- Trifecta is a red wine
- Beaucanon Estate Chardonnay is a white wine
- Beaucanon Estate Winery is located in California,
- etc.

The architecture of Semantic Web does not only include ontologies. In general, it is more complex and consists of:

- infrastructural levels: (UNICODE: character set, URI: Universal Resource Identifiers, XML: syntax, Cryptography: safety)
- taxonomies (RDF: Resource Description Framework, RDF Schema)
- ontologies (OWL: Web Ontology Language(s))
- rules (RIF, SWRL, ...)
- queries (SPARQL, ...)
- logic and reasoning (description logics, rule-based languages)
- trust (rating agencies/services providing information about trust level of web agents and web services)
- application level: user interfaces and applications.

In this article we mainly focus on logic and reasoning related to Semantic Web. The major logical formalisms for ontology specification and reasoning in Semantic Web are description logics, as discussed in the next section.

DESCRIPTION LOGICS

Description logics are a family of knowledge representation languages, with well defined semantics. They are used to represent terminological knowledge of a given application domain in a structured and formally well-understood way [4]. They are decidable fragments of first-order logic. In the context of Semantic Web, they form a firm foundation for ontology languages used for conceptual modeling. The logical formalism of the OWL 2 Web Ontology Language, recommended by W3C,[3] is based on the description logic SROIQ [7].

[2] See, e.g., http://www.w3.org/DesignIssues/Semantic.html.

[3] See http://www.w3.org/.

Description logics represent the domain of interest in terms of concepts, individuals, and roles. A concept is interpreted as a set of individuals, while a role is interpreted as a binary relation between individuals. A knowledge base is then a triple consisting of:

- RBox: role axioms, specifying properties of roles
- TBox: terminological axioms, allowing one to specify complex concepts on the basis of simpler ones
- ABox: assertions (facts) about individuals (objects) and roles.

Usually, there is a trade-off between expressiveness and complexity of logics. The main line of research on description logics is then to provide as expressive and at the same time as efficient as possible reasoning techniques, allowing one to address problems like satisfiability of knowledge bases, instance checking or concept inclusion. The paper [11][4] provides advanced reasoning techniques for a very expressive, but still decidable formalism which can serve as a basis for description logics.

Typical reasoning problems in the full version of OWL 2 are not tractable. This limits applicability of this language to ontologies with relatively simple relationships among concepts, allowing reasoning engines to focus on subontologies of a restricted, local scope. Therefore, OWL 2 language has additional profiles: OWL 2 EL, OWL 2 QL and OWL 2 RL, which are sublanguages of OWL 2 FULL with deterministic polynomial data complexity. They are based on the families of description logics EL [2, 3], DL-Lite [5] and DLP (Description Logic Programs) [6], respectively.

Description logics rarely address problems of inconsistent information or non-monotonic reasoning, but see [10, 12] and references there. Rule languages are a more natural tool to address such problems.

RULE-BASED LANGUAGES IN SEMANTIC WEB

Rule-based languages can be used as:

- ontology description languages
- user query languages.

Many such languages have been developed by restricting description logics to their suitably selected Horn fragments. Another idea depends on combining more traditional DATALOG-based languages [1] with description logics. However, as we attempt to show below, rule languages frequently suffice even without combining them with description logics.

Recall that DATALOG rules are of the form:[5]

$$H(X) :- B_1(X_1), B_2(X_2), \ldots, B_k(X_k).$$

In the above rule H, B_1 B_2, ..., B_k are names of relations and X, X_1, X_2, ..., X_k are tuples consisting of variables and constants. H(X) is called the *head* of the rule and $B_1(X_1)$, $B_2(X_2)$, ..., $B_k(X_k)$ is called the *body* of the rule.

The meaning of such rules is the following:

the conjunction of $B_1(X_1)$, $B_2(X_2)$, ..., $B_k(X_k)$ implies H(X)

or, in other words,

the intersection of sets of tuples satisfying $B_1(X_1)$, $B_2(X_2)$, ..., $B_k(X_k)$ is included in the set of tuples satisfying H(X).

Rule languages typically accept Close World Assumption, where facts not deducible from the underlying knowledge base are assumed to be false and can be used in further reasoning. Semantic Web accepts the Open World Assumption, where such non-deducible (unknown) facts must not be used by reasoners.

Of course, set-theoretical inclusion reflects the subconcept relationship. Therefore, typical taxonomies and ontology properties are easily expressible using rules.[6] For example, we can have the following rules (reflecting a TBox):

[4] Partially worked out when the author of the current paper has been a visiting professor in Warsaw School of Computer Science.

[5] There are some other restrictions, too – see [1]. Note also that negation is not allowed in DATALOG rules.

[6] Let us emphasize that some Description Logics constructs are provably inexpressible by such rules.

- wine(X):- redWine(X). -- red wine is a subconcept of the concept wine
- wine(X):- whiteWine(X). -- white wine is a subconcept of the concept wine
- wine(X):- desertWine(X). -- desert wine is a subconcept of the concept wine

Note that those rules, in the presence of Closed World Assumption correspond to the equality:

the set of wines = the union of the set of red wines, the set of white wines and the set of desert wines.

Accepting the Open World Assumption the above rules express subconcept relationship:

the set of wines includes the union of the set of red wines, the set of white wines and the set of desert wines.

We can easily formulate facts (reflecting an ABox):

- producedBy(trifecta, beaucanonEstateWinery).
  -- Trifecta is produced by Beaucanon Estate Winery
- redWine(trifecta). -- Trifecta is a red wine
- whiteWine(beaucanonEstateChardonnay).
  -- Beaucanon Estate Chardonnay is a white wine
- locatedIn(beaucanonEstateWinery, California).
  -- Beaucanon Estate Winery is located in California.

Observe that one can specify new concepts, like:

wanted(X):- redWine(X), producedBy(X,Y), locatedIn(Y, California).

The concept ‘wanted’ is intended to include objects one is interested about. In our case, the query wanted(X) should return information (or web pages including such an information) about red wines produced by a winery located in California. Such a query cannot be expressed by a list of keywords.

Rule languages mentioned above rarely address problems of inconsistent or incomplete information (but see [8] and references there). However, in the 4QL language, discussed in the next section, incomplete information and inconsistencies are first-class citizens.

RULE-BASED QUERY LANGUAGE 4QL

4QL [8, 9][7] is a general purpose query language as a language suitable for Semantic Web applications. It is:

- a rule-based database query language with negation allowed in bodies and heads of rules
- the first such language with tractable and at the same time intuitive semantics, even though the area of deductive databases is over 30 years old
- founded on a four-valued semantics with truth values: true, false, inconsistent and unknown.

In addition, it:

- supports a modular and layered architecture and provides a tractable framework for many forms of rule-based reasoning both monotonic and nonmonotonic
- is tractable w.r.t. data complexity and allows one to express all tractable queries
- deals with incomplete and inconsistent information, allowing users to reduce unknown and inconsistent zones by simple constructs, providing means for a uniform treatment of:
- Open and Local Closed World Assumption (thus Closed World Assumption, too)
- other nonmonotonic/commonsense formalisms, including various variants of default reasoning, autoepistemic reasoning and other formalisms
- application-specific disambiguation of inconsistent information, including defeasible reasoning.

The above features are achieved, among others, by such constructs as modules and external literals. Rules can be distributed among modules. To access relation ‘r’ defined in module ‘M’ we write M.r. Modules are placed into layers. External literals allow one to check truth values of *literals*, i.e., expressions of the form M.r(X) or -M.r(X), where ‘-’ stands for negation. An *external literal* takes the form L in S, where L is a literal and S is a set of truth values. For example,

[7] See also 4ql.org.

M.r(X) in {true, incons}

is true when the truth value of M.r(X) is ‘true’ or ‘inconsistent’. Whenever a module, say N, refers to a literal M.T, then M is required to be in a layer strictly lower than N.[8]

Consider the following example, based on the one given in [10] in the context of description logics. A web service M supplies information about stocks. A user U looks for low risk stocks, promising big gain. The user’s query can be expressed by:

U.int(X):- M.lr(X), M.bg(X).

In the above rule ‚int’, ‚lr’ and ‚bg’ stand for „interesting”, „low risk” and „big gain”, respectively. For simplicity, assume that the service M has a knowledge base consisting only of the following facts provided by different experts/agents/other services:

- M.lr(s1), -M.lr(s1), M.bg(s1)
- -M.lr(s2), -M.bg(s2)
- M.lr(s3)
- M.lr(s4), M.bg(s4).

When a set L of literals is given, the truth value of a given literal, say M.l, is:

- 'true' when M.l is in L and M.l is not in L
- 'false' when M.l is not in L and -M.l is in L
- 'incons' when both M.l and -M. are in L
- 'unknown' when none of M.l, -M.l is in L.

Returning to our example, we have that:

- M.lr(s1)='incons', M.lr(s2)='false', M.lr(s3)='true', M.lr(s4)='true'
- M.bg(s1)='true', M.bg(s2)='false', M.bg(s3)='unknown', M.bg(s4)='true'.
- According to the semantics of 4QL, the query U.int(X) computes truth values for all stocks which appear in the database, i.e., for s1, s2, s3 and s4:

U.int(s1)='incons', U.int(s2)='false', U.int(s3)='unknown', U.int(s4)='true'.

Now the user can express the desired reaction on received truth values, for example deciding to buy stocks for which 'int' is 'true' or 'incons' and not buying those for which 'int' is 'false' or 'unknown':

U1.buy(X) :- U.int(X) in {true, incons}.
-U1.buy(X):- U.int(X) in {false, unknown}.

The above rules result in:

- U1.buy(s1)='true', U1.buy(s4)='true'
- U1.buy(s2)='false', U1.buy(s3)='false'.

A PROBLEM

Rule-based approach enables one to express concepts, taxonomies and complex ontologies. However, in some applications ontology alignment and maintenance are necessary. Even though Sematic Web services and agents use different ontologies, they still have to communicate with one other. In such cases determining correspondences between concepts is crucial. In different ontologies the same concept may be named and defined differently. Consider, for example, two ontologies:

- the first one with the concept 'child'
- the second one with the concept 'parent'.

Now in the first ontology a 'grandparent' concept can be defined by

O1.grandparent(X,Y):- O1.child(Z,X), O1.child(Y,Z).

-- X is a grandparent of Y if there is Z being a child of X and having Y as a child.

In the second ontology the same concept can be defined by:

O2.gp(X,Y):- O2.parent(X,Z), O2.parent(Z,Y).

-- X is a grandparent of Y if there is Z such that X is a parent of Z and Z is a parent of Y.

The question is how to match concept O1.grandparent with O2.gp?

[8] This generalizes the notion of stratification in DATALOG programs with negation in bodies of rules [1].

There is an ongoing research on ontology alignment and there are software tools supporting this process. It seems, however, that the future is in standardization of ontologies and a rigorous user specification of all deviations from standards.

CONCLUSIONS

Modern search engines are based on keyword-based search with relatively low expressive power, approximately corresponding to Boolean queries. In this article we have discussed selected issues related to Semantic Web, proposed as a remedy to many problems making automated web information processing hard. We have concentrated on the role of ontologies and logical formalisms applied to specify, verify and reason about them. We have emphasized that such formalisms are much more expressive than keyword-based queries. Since information sources both in today's and Semantic Web typically provide incomplete information and may cause inconsistencies, we have mainly addressed problems related to these issues.

In Semantic Web description logics and rule languages are major logical tools used in reasoning and search. We have concentrated on rule languages and indicated that the 4QL query language can be a suitable choice for gathering knowledge from heterogeneous sources and enabling users to react on unknown or inconsistent information in a well-controlled manner.

REFERENCES

[1] Abiteboul S., Hull R., Vianu V., *Foundations of Databases*, Addison-Wesley Pub. Co., 1996

[2] Baader F., Brandt S., Lutz C., *Pushing the EL envelope*, in: *Proceedings of IJCAI'2005*, (eds.) Kaelbling L.P., Saffiotti A., Morgan-Kaufmann Publishers, 2005, pages 364-369

[3] Baader F., Brandt S., Lutz C., *Pushing the EL envelope further*, in: *Proc. of the Washington DC workshop on OWL: Experiences and Directions (OWLED08DC)*, 2008

[4] Baader F., Sattler U., *An overview of tableau algorithms for description logics*, „Studia Logica" 2001, Vol. 69, pages 5-40

[5] Calvanese D., De Giacomo G., Lembo D., Lenzerini M., Rosati R., *Tractable reasoning and efficient query answering in description logics: The L-Lite family*, „J. Autom. Reasoning" 2007, Vol. 39, No. 3, pages 385-429

[6] Grosof B.N., Horrocks I., Volz R., Decker S., *Description logic programs: combining logic programs with description logic*, in: WWW2003, Budapest, May 20-24, 2003, pages 48-57

[7] Horrocks I., Kutz O., Sattler U., *The even more irresistible SROIQ*, in: *Proceedings of KR'2006*, (eds.) Doherty P., Mylopoulos J., Welty C.A., AAAI Press, 2006, pages 57-67

[8] Małuszyński J., Szałas A., *Living with Inconsistency and Taming Nonmonotonicity*, in: *Datalog 2010*, (eds.) Gottlob G., Grasso G., Moor O. de, Sellers A., LNCS 6702, Springer-Verlag, 2011, pages 334-398

[9] Małuszyński J., Szałas A., *Logical Foundations and Complexity of 4QL, a Query Language with Unrestricted Negation*, „J. Applied Non-Classical Logics" 2011, Vol. 21, No. 2

[10] Nguyen L.A., Szałas A., *Three-Valued Paraconsistent Reasoning for Semantic Web Agents*, in: *KES-AMSTA* (1), LNCS 6070, Springer-Verlag, 2010, pages 152-162

[11] Nguyen L.A., Szałas A., *ExpTime Tableau Decision Procedures for Regular Grammar Logics with Converse*, „Studia Logica" 2011, Vol. 98. No. 3, pages 387-428

[12] Szałas A., *Second-order Reasoning in Description Logics*, „J. Applied Non-Classical Logics" 2006, Vol. 16, No. 3-4, pages 517-530