RDF and Inconsistency

Claudio Gutierrez, Carlos Hurtado

Universidad de Chile
Computer Science Department
Center for Web Research

www.ctw.cl
Contents

- Consistency issues
- RDFS and OWL
- RDF formal model
- The context

RDF and Inconsistency
- Deal with natural inconsistencies of such data
- Build tools to process such data
- Give structure to the information on the Web

• Database point of view:
  - Make information on the Web machine-processable
  - Give semantics to the information on the Web

Semantic Web

• The context
not attract much interest."

But in academic logic, these practical Leibnizian tasks do
Leibniz would be probably enthusiastic about this new arena of
[...]

The Web gives a completely new perspective to Leibniz’s project,

\[\text{A reflection}\]
Our focus

RDF and Inconsistency

codification, transport, etc.

RDF Schema

OWL

reasoning, proofs, etc.

...
relations, cardinality, equality, etc.

OWL: more vocabulary for describing properties and classes (e.g.,

RDFS: a vocabulary for describing properties and classes of RDF

between them. These data model can be represented in XMI

RDF: basic data model for objects (resources) and relations

RDF, RDFS and OWL
An RDF graph is a set of RDF triples.

\[(I \cap B \cap \Omega) \times \Omega \times (B \cap \Omega) \in o, p, s\]

**Definition**

\[\text{Infinite set of literals} = I\]

\[\text{Blank nodes} = \{N \in \ell : \ell q\} = B\]

\[\text{Infinite set of URI references} = \Omega\]
\( \emptyset = \emptyset \), \( G_2 \cong G_2 \)

The merge of two graphs \( G_1, G_2 \) is defined as \( G_1 \cup G_2 \), where

renaming its blank nodes consistently,

\( \cong G_2 \) (isomorphic, “equal”) if \( G_2 \) is obtained from \( G_1 \) by (called instance of \( G \))

\( o \in (o) \in (d) \in (s) \) such that \( s \) is the set of all \( (d) \) in \( (s) \)\n
Intervals

A mapping is a function \( n : nBL \rightarrow \text{UBL preserving unaries and} \)

RDF Formal Model (cont.)
Theorem 6. RDF Semantics, Interpolation Lemma

Let $G_1, G_2$ be RDF graphs. Then $G_1$ entails $G_2$ (denoted $G_1 \models G_2$) if and only if an instance of $G_2$ is a subgraph of $G_1$. Let there be a mapping $\mu$ such that $\mu(G_2) \subseteq G_1$. $G_1 \models G_2$ if and only if an instance of $G_2$ is a subgraph of $G_1$. 
\( X, \text{ TruthValue, False} \)
\( X, \text{ Object, Maria} \)
\( X, \text{ Predicate, Loves} \)
\( X, \text{ Subject, John} \)
\( X, \text{ Type, Statement} \)

Reification:

\( \forall cty, Tier \) (city, Tier)
\( \forall street, Goethe St. \) (street, Goethe St.)
\( \forall address, Maria \) (address, Maria)

Structured Properties:

\( \forall John, Loves, Maria \) (John loves Maria)

Descriptions:

RDF: Flexible model
Subgraph isomorphism problem:

Complexity: Deduction for RDF is NP-complete (Proof: co-NP complete) •

Order

Expressiveness: Fragment $\exists \forall \mathsf{stat}(x,y,z), c_1, c_2, \ldots$ of first

RDF: expressiveness/complexity
Versus references to the components of a statement.

Reification: statements are given identities.

Notation: \( C_1 \models C_2 \) if \( C_2 \) is isomorphic to a subgraph of \( C_1 \).

Although can simulate W3C.

View:

Embed RDF in F-Logic (Yang, Kifer). Some differences with W3C.

RDF: alt. formalization.
rdfl:rest The rest of the subject RDF list after the first item.
rdfl:rest The rest item in the subject RDF list.
rdfl:member A member of the subject container.
rdfl:comment A description of the subject resource.
rdfl:label A human-readable name for the subject.
rdfl:range A range of the subject property.
rdfl:domain A domain of the subject property.
rdfl:subPropertyOf The subject is a subproperty of a property.
rdfl:subclassOf The subject is a subclass of a class.
rdfl:type The subject is an instance of a class.

B. Properties:

Vocabulary: RDF Schema
rdfl:subject The subject of the subject RDF statement.
rdfl:predicat The predicate of the subject RDF statement.
rdfl:object The object of the subject RDF statement.
rdfl:value Literal or property used for structured values.
rdfl:definedBy The definition of the subject resource.
rdfl:seeAlso Further information about the subject resource.

Properties (cont.):
- Datatypes
- Class intersection: incompatible with, incompatible with backward-compatible
- Header Information: imports, priority
- cardinality
- Restricted cardinality: minCardinality, maxCardinality
- Property Character: inverseOf, transitiveProperty
- SymmetricProperty, FunctionalProperty
- InverseFunctionalProperty
- Property "Type Restrictions": allValuesFrom, someValuesFrom
- Ontology Web Language (OWL)

RDF and Inconsistency
Web

Challenge: tackle logical inconsistencies of information on the

Web

avoid axioms and predefined vocabulary

- General design philosophy: as simple as possible (in particular,

RDF gives a semantic layer to the web (base to reasoning)
- **Answer 3**: Yes, and no (inconsistent)

- **Answer 2**: No. But in any reasonable extension, yes.

- **Answer 1**: No, there is no negation.

Logical inconsistencies in RDF specifications
- In Amazon, Neruda's "20 Love Poems" has two different

Example:
- Two statements in different pages
- Two statements inside a page
- There are (at least) two kinds of conjunction on the Web:

Inconsistency needs negation + conjunction

Issue I: conjunction
comparative: "The SW needs two kinds of negation" C. Wagner

- logical counterpart: two kinds of conjunctions: \forall
- merge of graphs - graph
- two related concepts in RDF
- contradictory (two statements that forgetting its source are
- Disagreement (two contradictory statements made in the same source)
- Inconsistency (two contradictory concepts)
Case:
- Procedure: search for the "right" mechanism for particular KB makeups, preferences, etc.
- Idea: Build ontologies to deal with conceptual tools (logics)
- Goal: Work in the presence of inconsistencies
- Facts: consistency is an exception

Web setting:
- Procedure: implement the algorithms
- Idea: Use adequate logic and/or makeup you KB
- Goal: Avoid inconsistency
- Facts: inconsistency is an exception

Classical:

Issue II: What is to be done?
- Use preferences

- Do not use this intro in further reasoning

- Use catalog of trustworthy people

- Run hubs and authorities alg. on certain pages

Favorite method (ontology)

Look in Yellow Pages of anti-inconsistency tools and choose your

- Ted Bertossis Page is not trustworthy

- Ted Bertossis Page is trustworthy

Issue 1.2: example
Inconsistency and Description Logics

- Use standard machinery to deal with inconsistency in OWL:
- sameIndividuals and differentFrom
- FunctionalProperty and InverseFunctionalProperty

Not true in OWL, for example:
- General philosophy of RDF: avoid axioms that constrain the meaning of its vocabulary.

Issue III: Adding Constraints
Several levels of statements' paradoxes

Is it reasonable to have a predicate 'triple (o,d,s) belongs to uni

RDF statements must be somewhere located

(dynamic Web pages)

Sources of RDF data on the Web: Web pages, data sources

Issue 1.4: References
A "paradox":

\[ (\text{object}, \text{subject}, \text{type, statement}, \text{truth value}) \]

\[ (\text{object}, \text{subject}, \text{type, statement}) \]

\[ (\text{object}, \text{subject}, \text{type, statement}, \text{truth value}) \]

\[ (\text{object}, \text{subject}, \text{type, statement}) \]

\[ (\text{object}, \text{subject}, \text{type, statement}, \text{truth value}) \]

\[ (\text{object}, \text{subject}, \text{type, statement}) \]

RDF and Inconsistency

Issue IV: Example: a "paradox"
But, if we view RDF graphs as databases, it is natural to add minimal constraints at the data level.

Database: set of RDF graphs

Key difference: presence of blank nodes

RDF Graph = standard relational table

Idea: RDF Graph = database

Issue V: RDF graphs as databases
Database Issue: not (yet) deductive issue

What to do if we find two subjects for a statement?

\[ X \rightarrow X \text{ for } \text{stat}(X, \text{subject}, Y), \text{etc.} \]

plus functional dependencies like:

\[
\begin{align*}
\text{stat}(X, \text{object}, q) &\quad \Rightarrow (q, r, c) \\
\text{stat}(X, \text{predicate}, q) &\quad (q, r) \\
\text{stat}(X, \text{subject}, q) &\quad (q, r) \\
\end{align*}
\]

Constraints for Reification. Need axioms like:

**Issue V: Example**
(Ex. from Fikes, Hayes, Horrocks, Doh) count the number of Joe brothers. There is no consistent way to brothers, but no name is specified. Example: Joe belongs to the class of people having exactly 3 persons of the same name.

RDF. "aggregating relations" instead of aggregating functions. Not easy to define aggregation (even at elementary levels) in forms of processing available RDF into.

Not only issue of lost pages, non-accessible sites, but different aggregation as source of inconsistencies on the Web.

Issue VI: Aggregation
Thanks for your time!

Ideas, comments, pointers, very welcomed.