



UiO : **University of Oslo**

Conservative Repurposing RDF Data

Audun Stolpe and Martin G. Skjæveland

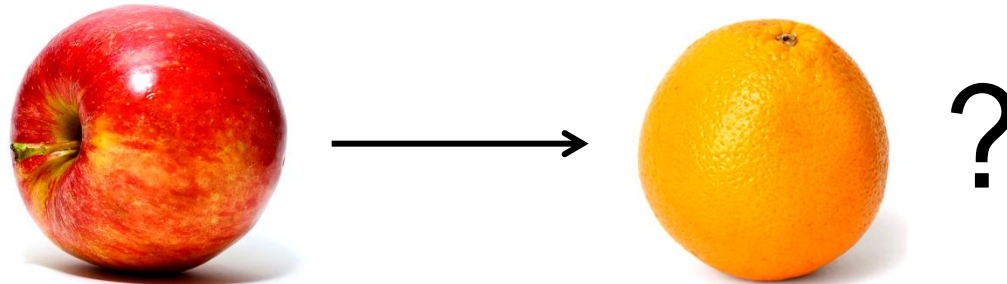
ISWC 2011 Minute Madness

25.10.2011



Motivation:

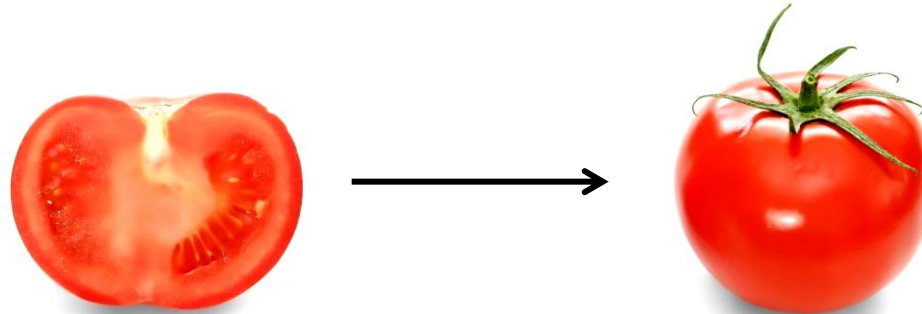
Transforming RDF data into different vocabulary



- “How do I make sure information is not lost?”
- “Can I get back my original data?”
- “Is it safe to merge with existing data?”
- “Can I add more information to the target?”

Solution:

Conservative RDF Transformations



- Homomorphism (1) with back-conditions (2):
 1. Systematic transformation of the source.
 2. Non-distortive representation of the source at the target.
- Different back-conditions give:
 - exact, merger or extending transformations.
- Applies also to SPARQL CONSTRUCT queries:
 - “Is my construct query a conservative transformation?”

Welcome!

- Please visit my poster stand.
- I have:
 - Poster.
 - Live demo.
 - Handouts.

Conservative Repurposing of RDF Data

Audun Stolpe and Martin G. Skjæveland, audun@iuhp.uio.no, mgs@iuhp.uio.no, Department of Informatics, University of Oslo

| Synopsis | Homomorphisms and bounds | Generatising from triples to chains | Example contd. |
|---|---|---|--|
| <p>Synopsis and non-destructive RDF interpretation.</p> <ul style="list-style-type: none"> - Change representation, preserve content. - Constructive cumulative error in relational data re-representation. - Handle exact, vague, and missing transformations. - Using homomorphisms with back-conditions. - Consider only RDF graph anchors, not resources. - Only RDF anchors may be transformed. - RDF anchors and edges may be annotated. - Applies to SPARQL CONSTRUCT queries. - Prototype with application available. | <p>Def. (RDF homomorphism). An RDF homomorphism \mathcal{H} from \mathcal{G}_1 to \mathcal{G}_2 is a function from \mathcal{E}_1 to \mathcal{E}_2 such that if \mathcal{E}_1 is a triple (p, s, o), then $\mathcal{H}(p, s, o)$ is a triple (p', s', o').</p> <p>Def. (Upper bound). A function f from a graph \mathcal{G}_1 to \mathcal{G}_2 is an RDF homomorphism from \mathcal{G}_1 to \mathcal{G}_2 if and only if f is an RDF homomorphism from \mathcal{G}_1 to \mathcal{G}_2.</p> <p>Def. (Bound). $\mathcal{H}_1 \rightarrow \mathcal{H}_2$ if \mathcal{H}_1 map to \mathcal{H}_2 is bounded by the following condition:</p> $\forall p, s, o \in \mathcal{E}_1, p', s', o' \in \mathcal{E}_2, \text{if } (p, s, o) \in \mathcal{E}_1 \text{ and } (p', s', o') \in \mathcal{E}_2 \text{ then } (p, s, o) \in \mathcal{H}_1 \rightarrow \mathcal{H}_2(p, s, o) \rightarrow (p', s', o')$ <p>We will use the notation $\mathcal{H} \rightarrow \mathcal{H}'$ to denote that $\mathcal{H} \rightarrow \mathcal{H}'$ is bounded.</p> | <p>Change generative triple to triple → map to chain to chain construction.</p> <p>The graph \mathcal{G}_1 contains a triple representation for each triple in \mathcal{G}_2, and \mathcal{H}_1 is the function which takes a chain c of \mathcal{G}_1 and maps it to c'. \mathcal{H}_1 is a strongly typed subset of \mathcal{H}_2. Considered from \mathcal{G}_1 to \mathcal{G}_2, \mathcal{H}_1 is a strongly bounded map.</p> $\mathcal{H}_1(c) = \mathcal{H}_2(\mathcal{H}_1(c))$ | <p>Let \mathcal{G} contain data on the same format as \mathcal{G}_1, but for a different set of triples. We want to transform and merge \mathcal{G} with \mathcal{G}_1. Now consider the transformation \mathcal{H} using the same \mathcal{G}_1. Then the address and geographical location anchors are merged under homomorphism, allowing the triple using old edges to be added, but ensuring that the main instance of \mathcal{G}_1 and \mathcal{G} are not merged.</p> |
| <p>Preliminaries</p> <p>Let \mathcal{G} denote the set of resources, blank nodes and literals.</p> <p>An RDF graph \mathcal{G} (or \mathcal{R}) is a set of triples, written $\langle \mathcal{E}, \mathcal{A} \rangle$, where $\mathcal{E} \subseteq \mathcal{G} \times \mathcal{G} \times \mathcal{G}$. We use graph homomorphisms and void labels subjects (s) and objects (o) collectively anchors and predicates (p) edges. The set of anchors and the set of edges of a graph need not be disjoint. Let \mathcal{E} denote the set of anchors and edges (s, p, o) and \mathcal{A} the set of anchors.</p> <p>We consider only the least general non-fragrant SPARQL. The answer to a SPARQL SELECT query is a set of chains of anchors from \mathcal{G}. A <i>chain</i> is a CONSTRUCT query written $\langle \mathcal{E}, \mathcal{A} \rangle$, where \mathcal{E} is a CONSTRUCT query and \mathcal{A} is the anchor data. The answer to a SPARQL CONSTRUCT query over an RDF graph \mathcal{G} is either $\langle \mathcal{E}, \mathcal{A} \rangle$ or \mathcal{R} graph.</p> <p>See poster paper (link in footer) for details and references.</p> | <p>Results</p> <p>A strongly bounded transformation $\mathcal{H}: \mathcal{G}_1 \rightarrow \mathcal{G}_2$ returns that a SELECT query Q over \mathcal{G}_1 returns the exact same result as the query Q' over \mathcal{G}_2 if the queries conform:</p> $\mathcal{G}_1 \xrightarrow{\mathcal{H}} \mathcal{G}_2$ $Q \xrightarrow{\mathcal{H}} Q'$ <p>Similar results apply for INSERT and DELETE.</p> <ul style="list-style-type: none"> - Strongly typed do not allow any "lost" anchors in the target to be added by "old" edges. - Limited inputs allow old anchors and only to be related by old edges. - Input → output edges may not be deleted to be related by old edges. - They may delete old anchors but not to be related by old edges in case maps. <p>A comparison of the constructed → map preserves that retained bound → maps bounded constructive error in the set of anchors (missing).</p> <p>Standard maps: Edge may be transformed under off-anchors.</p> | <p>Example</p> <p>Let \mathcal{G} contain data about culturally valuable buildings in Oslo as anchored:</p> <pre> SELECT ?s ?p ?o FROM \mathcal{G} WHERE ?s ?p ?o . </pre> <p>Consider \mathcal{H} to be a map \mathcal{H} using the vCard vocabulary with the CONSTRUCT query Q':</p> <pre> CONSTRUCT { ?s ?p ?o ?t ?d WHERE ?s ?p ?o ?t ?d FROM \mathcal{G} WHERE ?s ?p ?o ?t ?d . </pre> <p>The query represents a conservative re-purposing of \mathcal{G}. What the main reader of \mathcal{G}, address and geographical location data, is mapped with a strong bound, e.g., all anchors → anchors → are mapped to the main.</p> <p>\mathcal{H}_1 is a strongly typed subset of \mathcal{H}_2, ensuring that the data is strictly preserved in \mathcal{H}_1. "General purpose" vocabulary anchors → anchors → are not lost, allowing the main instance, allowing one to add types to data in the target (e.g., vCard data has location, because for the building type).</p> | <p>SPARQL CONSTRUCT</p> <p>The anchors and void labels of graphs SPARQL CONSTRUCT queries on the RDF graphs. Extended syntax to the SPARQL query language on anchors and void labels. It is a CONSTRUCT query which anchors on anchors or edges. It is a map from \mathcal{G}_1 to \mathcal{G}_2 bounded by one of \mathcal{G}_1, then \mathcal{H} is a map under the same bound from \mathcal{G}_1 to \mathcal{G}_2.</p> |
| | | | <p>Implementation: Mapper Dan</p> <p>Implementation features:</p> <ul style="list-style-type: none"> - Good: There exists an algorithm for identifying common graphs. - Not so good: When querying for → maps, concerning the comparison of a graph may exponentially increase in size. <p>Mapper Dan:</p> <ul style="list-style-type: none"> - Web app, identifying triple between graphs. - Fetches two RDF graphs or one CONSTRUCT query. - Finds maps according to user-defined requirements. - Suggests those found if necessary. - Can apply this to graph → - Has CONSTRUCT query from map, or - create queries with map. <p>• URL: http://iuhp.uio.no/mapper/</p> |