Structural Summarization of Semantic Graphs
Ioana Manolescu

To cite this version:
Ioana Manolescu. Structural Summarization of Semantic Graphs. Extended Semantic Web Conference (ESWC), Jun 2018, Heraklion, Greece. hal-01808737

HAL Id: hal-01808737
https://inria.hal.science/hal-01808737
Submitted on 6 Jun 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Structural Summarization of Semantic Graphs

Ioana Manolescu

Head of CEDAR team
INRIA and Ecole Polytechnique, France
ioana.manolescu@inria.fr
http://pages.saclay.inria.fr/Ioana.Manolescu

ESWC Conference, June 5, 2018
Outline

1. **Motivation**: data discovery in semantic-rich RDF graphs
2. **Framework**: quotient summaries
   - Smaller graph which represents the original one in some sense
3. **Proposal**: use property cliques for summarizing explicit and implicit data
4. **Summarization algorithms**
5. **Perspectives**

Joint work with François Goasdoué (U. Rennes 1 and Inria), Paweł Guzewicz (U. Paris Saclay and Inria), and Šejla Čebirić (U. Paris Saclay and Inria) [ČGM15a, ČGM15b, ČGM17a, GM18, PGA⁺18]
Part I

Motivation: data discovery in RDF graphs
Big Data needs semantics

AI Magazine, Spring 2015
RDF graph discovery

An RDF graph can be large and complex, lack a fixed schema, include many heterogeneous values...
RDF summaries

**Simplified views** of an RDF graph [ČGK+18]

- Most often, a summary is also a **graph**, and/or: **statistics**, **patterns**...
- Summarize: the **data**, the **ontology**, **both**
- Many prior works on graph summarization applied to RDF
**Simplified views** of an RDF graph [ČGK⁺18]

- Most often, a summary is also a **graph**, and/or: **statistics**, **patterns**...
- Summarize: the **data**, the **ontology**, **both**
- Many prior works on graph summarization applied to RDF

Summary uses:

1. **For query processing**: give direct access to a group of nodes summarized together, detect empty queries...
2. **For data discovery**: help identify interesting structure or patterns in the data
RDF graphs are often structurally heterogeneous

Subject types in DBLP bibliographic data:

Type distribution (Click All Subjects or a certain type below for further exploration.)
RDF graphs are often structurally heterogeneous

Data properties of DBLP conference articles:
Our goal

Define **structural summaries** which:

- resist to **heterogeneity**
- flexibly take into account **RDF types**
- allow summarizing **implicit triples**
- can be **implemented efficiently**
The Resource Description Framework (RDF)

RDF graph: set of triples

```
Class
    resource (URI)
    blank node
    "literal (string)"
    property

"1949"
    publishedIn

doi1
    hasTitle
    "El Aleph"

Book
    rdf:type

_:b1
    hasName
    "J. L. Borges"
```
RDF Schema

We consider **RDFS** deductive constraints, stating connections between classes and properties

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Triple</th>
<th>OWA interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Subclass</strong></td>
<td>$c_1 \text{ rdfs:subClassOf } c_2$</td>
<td>$c_1 \subseteq c_2$</td>
</tr>
<tr>
<td><strong>Subproperty</strong></td>
<td>$p_1 \text{ rdfs:subPropertyOf } p_2$</td>
<td>$p_1 \subseteq p_2$</td>
</tr>
<tr>
<td><strong>Domain typing</strong></td>
<td>$p \text{ rdfs:domain } c$</td>
<td>$\Pi_{\text{domain}(p)} \subseteq c$</td>
</tr>
<tr>
<td><strong>Range typing</strong></td>
<td>$p \text{ rdfs:range } c$</td>
<td>$\Pi_{\text{range}(p)} \subseteq c$</td>
</tr>
</tbody>
</table>

“Any $c_1$ is also a $c_2$”
RDF Schema

Simple language of **deductive constraints** between classes and properties

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Triple</th>
<th>OWA interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subclass</td>
<td>( c_1 ) rdfs:subClassOf ( c_2 )</td>
<td>( c_1 \subseteq c_2 )</td>
</tr>
<tr>
<td>Subproperty</td>
<td>( p_1 ) rdfs:subPropertyOf ( p_2 )</td>
<td>( p_1 \subseteq p_2 )</td>
</tr>
<tr>
<td>Domain typing</td>
<td>( p ) rdfs:domain ( c )</td>
<td>( \Pi_{\text{domain}}(p) \subseteq c )</td>
</tr>
<tr>
<td>Range typing</td>
<td>( p ) rdfs:range ( c )</td>
<td>( \Pi_{\text{range}}(p) \subseteq c )</td>
</tr>
</tbody>
</table>

"If two resources are related by \( p_1 \), they are also related by \( p_2 \)"
RDF Schema

Simple language of **deductive constraints** between classes and properties

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Triple</th>
<th>OWA interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subclass</td>
<td>( c_1 \text{ rdfs:subClassOf } c_2 )</td>
<td>( c_1 \subseteq c_2 )</td>
</tr>
<tr>
<td>Subproperty</td>
<td>( p_1 \text{ rdfs:subPropertyOf } p_2 )</td>
<td>( p_1 \subseteq p_2 )</td>
</tr>
<tr>
<td>Domain typing</td>
<td>( p \text{ rdfs:domain } c )</td>
<td>( \Pi_{\text{domain}}(p) \subseteq c )</td>
</tr>
<tr>
<td>Range typing</td>
<td>( p \text{ rdfs:range } c )</td>
<td>( \Pi_{\text{range}}(p) \subseteq c )</td>
</tr>
</tbody>
</table>

“Anyone having \( p \) is a \( c \)”
RDF Schema

Simple language of **deductive constraints** between classes and properties

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Triple</th>
<th>OWA interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subclass</td>
<td>$c_1 \text{ rdfs:subClassOf } c_2$</td>
<td>$c_1 \subseteq c_2$</td>
</tr>
<tr>
<td>Subproperty</td>
<td>$p_1 \text{ rdfs:subPropertyOf } p_2$</td>
<td>$p_1 \subseteq p_2$</td>
</tr>
<tr>
<td>Domain typing</td>
<td>$p \text{ rdfs:domain } c$</td>
<td>$\Pi_{\text{domain}}(p) \subseteq c$</td>
</tr>
<tr>
<td>Range typing</td>
<td>$p \text{ rdfs:range } c$</td>
<td>$\Pi_{\text{range}}(p) \subseteq c$</td>
</tr>
</tbody>
</table>

“Anyone who is a value of $p$ is a $c$”
Open-world assumption and RDF entailment

RDF data model based on the open-world assumption.

Deductive constraints lead to **implicit triples**: part of the graph even though not explicitly present.
Open-world assumption and RDF entailment

RDF data model based on the open-world assumption.

Deductive constraints lead to **implicit triples**: part of the graph even though not explicitly present

\[
\text{explicit triples} + \text{entailment rules} \rightarrow \text{implicit triples}
\]
Open-world assumption and RDF entailment

RDF data model based on the open-world assumption.

Deductive constraints lead to **implicit triples**: part of the graph even though not explicitly present

$$\text{explicit triples} + \text{entailment rules} \rightarrow \text{implicit triples}$$

Exhaustive application of entailment leads to **saturation** (closure)
The semantics of an RDF graph $G$ is its saturation $G^\infty$.

RDF data graph and RDF schema graph:

Saturation of the graph union:
Part III

RDF summarization
RDF summaries

Problem
RDF graph $G$ is large, heterogeneous, partially implicit. How to compactly represent all its structure?

Existing solutions
**Partial** representation (frequent patterns, statistics etc.)
e.g., [NM11, LYL13]
**Potentially not compact** e.g., [GW97, CFKP15]
Only for **explicit data**, e.g., [CDT13, ZDYZ14]
A summary of DBLP data

150M triples
French territory division in regions, departments, urban areas, cities, districts etc.

368K triples
Summarization principle: quotient graphs

Let $\equiv$ be an equivalence relation on the nodes of $G$. The quotient $G_{\equiv}$ of a directed graph $G$ by $\equiv$ is a graph defined as follows:

- $G_{\equiv}$ nodes: one for $\equiv$ equivalence class of $V$
- $G_{\equiv}$ edges: $n_1^{\equiv} \overset{a}{\rightarrow} n_2^{\equiv}$ iff $\exists n_1 \overset{a}{\rightarrow} n_2 \in G$ such that $n_1$ represented by $n_1^{\equiv}$, $n_2$ represented by $n_2^{\equiv}$
Summarization principle: quotient graphs

Let \( \equiv \) be an equivalence relation on the nodes of \( G \). 

The quotient \( G_\equiv \) of a directed graph \( G \) by \( \equiv \) is a graph defined as follows:

- **\( G_\equiv \) nodes:** one for \( \equiv \) equivalence class of \( V \)
- **\( G_\equiv \) edges:** \( n_1^\equiv \overset{a}{\rightarrow} n_2^\equiv \) iff \( \exists n_1 \overset{a}{\rightarrow} n_2 \in G \) such that \( n_1 \) represented by \( n_1^\equiv \), \( n_2 \) represented by \( n_2^\equiv \)

Quotients have interesting summary qualities:

1. **Property completeness:** All \( G \) properties appear in \( G_\equiv \)
2. **Size guarantees:** By definition, \( G_\equiv \) is at most as large as \( G \) (usually much smaller)
3. **Structure representativeness:** Given a query \( q \), if its structure-only version is empty on \( G_\equiv \), then \( q \) is empty on \( G \)
Common graph quotients: bisimilarity [HHK95]

Two nodes are forward (resp. backward) bisimilar if they have exactly the same incoming (resp. outgoing) paths; $\sim_{fw}$, $\sim_{bw}$, $\sim_{fb}$
Common graph quotients: bisimilarity [HHK95]

Two nodes are forward (resp. backward) bisimilar if they have exactly the same incoming (resp. outgoing) paths; \( \sim_{fw}, \sim_{bw}, \sim_{fb} \)

**Problem:** Bisimilarity compresses/summarizes very little!
Common graph quotients: bisimilarity [HHK95]

Two nodes are forward (resp. backward) bisimilar if they have exactly the same incoming (resp. outgoing) paths; \( \sim_{fw}, \sim_{bw}, \sim_{fb} \)

**Problem:** Bisimilarity compresses/summarizes very little!

**Solution:** Bounded bisimilarity, e.g., \( \sim_{1fb} \) (the weakest)
Two nodes are forward (resp. backward) bisimilar if they have exactly the same incoming (resp. outgoing) paths; \( \sim_{fw}, \sim_{bw}, \sim_{fb} \)

**Problem:** Bisimilarity compresses/summarizes very little!

**Solution:** Bounded bisimilarity, e.g., \( \sim_{1fb} \) (the weakest)

Still: > 130 property combinations on conf. papers in DBLP
Common graph quotients: bisimilarity [HHK95]

Two nodes are forward (resp. backward) bisimilar if they have exactly the same incoming (resp. outgoing) paths; \( \sim_{fw}, \sim_{bw}, \sim_{fb} \)

**Problem:** Bisimilarity compresses/summarizes very little!

**Solution:** Bounded bisimilarity, e.g., \( \sim_{1fb} \) (the weakest)

Still: > 130 property combinations on conf. papers in DBLP

**Requirement 1:** We need equivalence relationships robust to structural heterogeneity.
What about type and schema triples?

Can we apply quotientization directly to an RDF graph?
What about type and schema triples?

Can we apply quotientization directly to an RDF graph? Sample graph $G$ and a possible quotient:

$A$ type $u_1$ $n_2$ type $B$

$p$ $p$ $p$ $r$ $p_1$ $p_1$ $p$ $r$

$\Rightarrow$

$p$ $r$

$p$

$r$

$\Rightarrow$

$\bigcirc$

$\bigcirc$
What about type and schema triples?

Can we apply quotientization directly to an RDF graph?
Sample graph $G$ and a possible quotient:

Possible loss of class and property names
What about type and schema triples?

Can we apply quotientization directly to an RDF graph?
Sample graph $G$ and a possible quotient:

Possible loss of schema triples
What about type and schema triples?

Can we apply quotientization directly to an RDF graph?
Sample graph $G$ and a possible quotient thereof:

$$
\begin{align*}
A \xleftarrow{\text{type}} & u_1 \quad n_2 \xrightarrow{\text{type}} B \\
& \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \ quad
What about type and schema triples?

Can we apply quotientization directly to an RDF graph? Sample graph $G$ and a possible quotient thereof:

Possible loss of implicit triples

**Requirement 2:** Avoid schema loss through quotientization
RDF equivalence relation and RDF summaries

To meet Requirement 2, we define:

1. **RDF equivalence relation**: an equivalence relation on RDF graph nodes such that any class or property node is only equivalent to itself.

2. **RDF summary**: a quotient of a graph $G$ by an RDF equivalence relation such that any class or property node is represented by itself.

**Consequence**: For any RDF equivalence relation $\equiv$ and RDF graph $G$, the schema of $G/\equiv$ is the schema of $G$.

$\Rightarrow$ No schema compression! (to be rediscussed briefly)
Summarization through an RDF equivalence relation

E.g., let $\equiv_{1fb}$ to be the RDF node equivalence obtained from $\sim_{1fb}$. Sample graph $G$:

Its quotient through the RDF node equivalence $\equiv_{1fb}$:
Recap

We have seen:

- **RDF node equivalence** and **RDF quotients** ⇒ structural representativeness, empty query pruning

We still need to solve:

- **Requirement 1**: compact (yet preserve structure) even on heterogeneous graphs
- **Requirement 3**: can we summarize implicit triples?

We will address them in this order.
RDF node equivalence based on property cliques

Intuition: $n_1, n_2$ are “of the same kind”; similarly $b_1, b_2, b_3$

$n_3, n_4$ may or may not be of the same kind as $n_1, n_2$. 

Clique-based summarization

RDF node equivalence based on property cliques

Output property cliques: \{a, b, d\}; \{f\}; \{g\}; ∅

Input property cliques: \{a\}; \{b\}; \{d\}; \{f\}; \{g\}; ∅
Weak clique-based summaries

Two nodes are weakly equivalent \((\equiv_W)\) iff they have the same input clique or the same output clique or are weakly equivalent to a third one.

Weak summary \(G/W\) of the sample RDF graph \(G\):

\[
\begin{align*}
n_1 & \quad n_2 & \quad n_3 & \quad n_4 & \quad n_5 & \quad n_6 \\
a_1 & \quad b_1 & \quad b_2 & \quad d_1 & \quad a_2 & \quad b_3 & \quad b_4 & \quad d_2 \\
a & \quad b & \quad b & \quad d & \quad a & \quad b & \quad b & \quad d \\
f & \quad g \\
\Rightarrow \\
a & \quad b & \quad d \\
\end{align*}
\]
Weak clique-based summaries

Two nodes are weakly equivalent \((\equiv/W)\) iff they have the same input clique or the same output clique or are weakly equivalent to a third one.

Weak summary \(G/W\) of the sample RDF graph \(G\):

Property: In \(G/W\), each data property appears exactly once \(\Rightarrow\) its nodes are “source of \(p\), target of \(p\)” for each \(p\) [ČGM15b].
Weak clique-based summaries

Property: $G/W$ nodes are “source of $p$, target of $p$” for each $p$.

Detecting errors in the data: why do the birthplace and deathplace loop?

Looking in the data, we find:

\[
\langle \text{http://dbpedia.org/resource/Kunitomo_Ikkansai} \rangle \langle \text{http://www.w3.org/1999/02/22-rdf-syntax-ns#type} \rangle \\
\langle \text{http://xmlns.com/foaf/0.1/Person} \rangle.
\]

\[
\langle \text{http://dbpedia.org/resource/Kunitomo_Ikkansai} \rangle \langle \text{http://dbpedia.org/ontology/birthPlace} \rangle \\
\langle \text{http://dbpedia.org/resource/Kunitomo_Ikkansai} \rangle.
\]
Strong clique-based summaries

Two nodes are strongly equivalent ($\equiv_{S}$) iff they have the same input clique and the same output clique.

Strong summary $G/\equiv_{S}$ of the same $G$:
Which role should node types play in summarization?

Having the same type(s) is orthogonal w.r.t. having the same structure.
Which role should node types play in summarization?

Having the same type(s) is orthogonal w.r.t. having the same structure. Two alternatives:

1. **Data-then-type**: group nodes first by their data triples, then carry the types from each equivalent group to its representative.
Which role should node types play in summarization?

Having the same type(s) is orthogonal w.r.t. having the same structure. Two alternatives:

- **Data-then-type:** group nodes first by their data triples, then carry the types from each group to its representative.

Extended Weak summary:
Adding types after data summarization

**Data-then-type:** group nodes first by their data triples, then carry the types from each group to its representative.

Extended Strong summary:
**Type-then-data:** Group nodes by their type set, and untyped nodes by their data properties.

Typed Weak summary $G/\equiv_{TW}$ of the sample graph:
Giving prominence to types

**Type-then-data:** Group nodes **first by their types.** Only untyped nodes are grouped by their data properties.

**Typed Strong** summary \( G/\equiv TS \) of the sample graph:
## RDF summaries outline

<table>
<thead>
<tr>
<th>Summary</th>
<th>Weak?</th>
<th>Strong?</th>
<th>Types first?</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G/\equiv W$</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$G/\equiv S$</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>$G/\equiv TW$</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>$G/\equiv TS$</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
## RDF summaries outline

<table>
<thead>
<tr>
<th>Summary</th>
<th>Weak?</th>
<th>Strong?</th>
<th>FW bisim?</th>
<th>BW bisim?</th>
<th>Types first?</th>
</tr>
</thead>
<tbody>
<tr>
<td>G/$\equiv$ W</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G/$\equiv$ S</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G/$\equiv$ TW</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>G/$\equiv$ TS</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>G/$\equiv$ fw</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G/$\equiv$ bw</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>G/$\equiv$ fb</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>G/$\equiv$ fw,T</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>G/$\equiv$ bw,T</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>G/$\equiv$ fb,T</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Relations between RDF summaries [ČGM17b]
## Summary size comparison (more in [ČGM17b])

| Graph $G$ | $|G|$   | Summary $G/\equiv$ | $|G/\equiv|$ | $cf_{\equiv}$ |
|-----------|--------|--------------------|--------------|--------------|
| DBLP      | 150,787,464 | $G/W$             | 71           | 2,123,767    |
| DBLP      | 150,787,464 | $G/S$             | 206          | 731,978      |
| DBLP      | 150,787,464 | $G/fw$            | 262,695      | 574          |
| LUBM1M    | 1,227,868   | $G/W$             | 161          | 7,579        |
| LUBM1M    | 1,227,868   | $G/S$             | 207          | 5,903        |
| LUBM1M    | 1,227,868   | $G/fw$            | 1982         | 617          |
| LUBM10M   | 11,990,183  | $G/W$             | 162          | 74,013       |
| LUBM10M   | 11,990,183  | $G/S$             | 206          | 58,204       |
| LUBM10M   | 11,990,183  | $G/fw$            | 24,958       | 480          |
| LUBM10M   | 11,990,183  | $G/bw$            | 6,162        | 1,944        |
| LUBM10M   | 11,990,183  | $G/fb$            | 11,990,076   | 1            |
Summarizing $G^\infty$

Recall: With an RDF Schema, the semantics of $G$ is $G^\infty \Rightarrow$
We really need $(G^\infty)/\equiv$!

1. Saturate $G$, then summarize
2. Can we avoid saturating $G$?...
Summarizing $G^\infty$

Recall: With an RDF Schema, the semantics of $G$ is $G^\infty \Rightarrow$
We really need $(G^\infty)/\equiv$!

1. Saturate $G$, then summarize
2. Can we avoid saturating $G$?...

**Shortcut theorem** [ˇCGM17a]

For the summaries $G/\equiv W$, $G/\equiv S$, $G/\equiv fw$, $G/\equiv bw$, $G/\equiv fb$:

$$(G^\infty)/\equiv$$ is the same as $$((G/\equiv)^\infty)/\equiv$$

Also: **sufficient condition** for any $\equiv$ to admit the shortcut.
Shortcut path to $G^\infty$

<table>
<thead>
<tr>
<th>Direct</th>
<th>$G \rightarrow \text{sat.} \rightarrow G^\infty \rightarrow \text{summ.} \rightarrow (G^\infty)_\equiv$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shortcut</td>
<td>$G \rightarrow \text{summ.} \rightarrow G_\equiv \rightarrow \text{sat.} \rightarrow (G_\equiv)^\infty \rightarrow \text{summ.} \rightarrow ((G_\equiv)^\infty)_\equiv$</td>
</tr>
</tbody>
</table>

If $G_\equiv$ is much smaller than $G$, **the shortcut may be faster!**

Up to 20 times in our experiments [ČGM17b]
Shortcut example: $G / \equiv W$
Shortcut counter-example: $G/\equiv TW$
Summarization algorithms

1. **Global algorithms**: visit all $G$, compute $\equiv$ relation, then traverse $G$ again and represent each triple in $G/\equiv W$.

2. **Incremental algorithms**: visit $G$, compute $\equiv$ and summary based on knowledge gained so far; adjust summary.

We devised global and incremental summarization algorithms for $G/\equiv W$, $G/\equiv S$, $G/\equiv TW$, $G/\equiv TS$.

Difficulty of incremental summarization: adjusting $\equiv$ and revisiting summarization decisions.
Example: weak incremental summarization (1)

Each color corresponds to a different $\equiv_w$ class
Example: weak incremental summarization (1)

Each color corresponds to a different $\equiv_w$ class
Example: weak incremental summarization (1)

Each color corresponds to a different $\equiv_w$ class
Example: weak incremental summarization (1)

Each color corresponds to a different $\equiv_w$ class
Example: weak incremental summarization (1)

Each color corresponds to a different $\equiv_w$ class
Example: weak incremental summarization (1)

Each color corresponds to a different $\equiv_w$ class
Example: weak incremental summarization (1)

Each color corresponds to a different $\equiv_w$ class.
Example: weak incremental summarization (2)
Example: weak incremental summarization (2)
Example: weak incremental summarization (2)
Example: weak incremental summarization (3)
Full graph and its summary:
Algorithm scale-up

$10^5$ to $1.5 \times 10^8$ triples
Summary-enabled LOD cloud exploration [PGA+18]

Collaboration with ILDA Inria data visualization team on LODAtlas
http://lodatlas.lri.fr/
Use summary to derive visualisation instead of the original graph
(smaller, faster)

abs-linked-data : Australian Bureau of Statistics (ABS) Linked Data
Part IV

Conclusion
The need for RDF graph discovery tools

- RDF graphs can be **large and complex**, they lack a prescriptive schema
- Semantic rules lead to **implicit data**
- **Structural quotient summaries** compactly represent graph structure and semantics.

Available online at: (new version soon)

https://team.inria.fr/cedar/projects/rdfsummary/

- Type-first summarization variant to cope with large type hierarchies [GM18]
- Integration into LODAtlas platform [PGA⁺18]
Part V

Perspectives
Ongoing and future work

Ongoing:
1. Experiment with new, parallel summarization algorithms based on Spark
2. Keyword search in RDF graphs based on quotient summaries

Future:
1. Controlled inclusion of data value synopsis in the summary
2. Extension to more expressive ontology languages
3. Integration in a larger platform for summary-based data discovery (with Mirjana Mazuran)
4. Exploration of interesting aggregate view of RDF graphs (with Yanlei Diao)
References


