A Framework for Efficient Representative Summarization of RDF Graphs

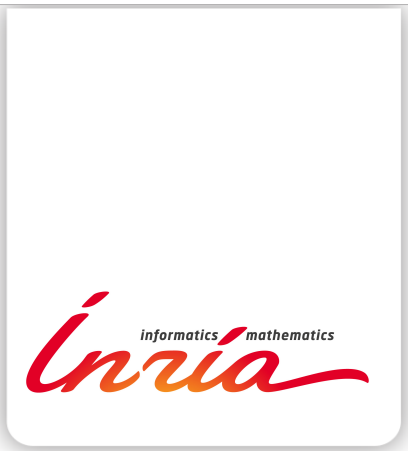
Šejla Čebirić, François Goasdoué, Ioana Manolescu

A Framework for Efficient Representative Summarization of RDF Graphs

Šejla Čebirić, François Goasdoué, Ioana Manolescu

RESEARCH REPORT
N° 9090
August 2017
Project-Teams CEDAR

ISSN 0249-6399 ISRN INRIA/RR--9090--FR+ENG



A Framework for Efficient Representative Summarization of RDF Graphs

Šejla Čebirić, François Goasdoué, Ioana Manolescu

Project-Teams CEDAR

Research Report n° 9090 — version 1 — initial version August 2017 — revised version Août 2017 — 11 pages

Abstract:
RDF is the data model of choice for Semantic Web applications. RDF graphs are often large and have heterogeneous, complex structure. Graph summaries are compact structures computed from the input graph; they are typically used to simplify users' experience and to speed up graph processing.
We introduce a formal RDF summarization framework, based on graph quotients and RDF node equivalence; our framework can be instantiated with many such equivalence relations. We show that our summaries represent the structure and semantics of the input graph, and establish a sufficient condition on the RDF equivalence relation which ensures that a graph can be summarized more efficiently, without materializing its implicit triples.

Key-words: Semantic Web, RDF, data summary, inference, reasoning, data compression

RESEARCH CENTRE
SACLAY – ÎLE-DE-FRANCE

1 rue Honoré d'Estienne d'Orves
Bâtiment Alan Turing
Campus de l'École Polytechnique
91120 Palaiseau

Une approche pour la construction efficace des résumés représentatifs de graphes RDF

Résumé : Le modèle RDF est très largement employé dans des applications du Web Sémantique. Les graphes de données RDF sont souvent grands et leur structure est complexe et hétérogène. Les résumés de graphes sont des structures compactes calculées à partir de tels graphes de données; ils sont employés pour faciliter l'interaction avec les grands graphes de données et afin de rendre leur traitement plus efficace.

Nous présentons une approche formelle de résumé de graphes RDF, basées sur les graphes quotient et sur une nouvelle notion d'équivalence de noeuds RDF; notre approche peut être instanciée avec de nombruses relations d'équivalence. Nous montrons nos résumés représentent la structure et la sémantique des graphes d'entrée, et établissons une condition suffisante sur la relation d'équivalence RDF pour que le résumé d'un graphe puisse être construit de façon efficace, sans matérialiser ses triples implicites.

Mots-clés : Web Sémantique, RDF, résumé de données, inférence, raisonnement, compression de données

RDF statement	Triple	Shorthand
Class assertion	$(s, \text{rdf:type}, o)$	(s, τ, o)
Property assertion	(s, p, o) with $p \neq \text{rdf:type}$	(s, p, o)

RDFS statement	Triple	Shorthand
Subclass	$(s, \text{rdfs:subClassOf}, o)$	$(s, \prec_{sc}, o)$
Subproperty	$(s, \text{rdfs:subPropertyOf}, o)$	$(s, \prec_{sp}, o)$
Domain typing	$(s, \text{rdfs:domain}, o)$	$(s, \hookleftarrow_d, o)$
Range typing	$(s, \text{rdfs:range}, o)$	$(s, \hookrightarrow_r, o)$

Name	Entailment rule
rdfs2	$(p, \hookleftarrow_d, o), (b_{s_1}, p, o_1) \rightarrow (b_{s_1}, \tau, o)$
rdfs3	$(p, \hookrightarrow_r, o), (b_{s_1}, p, o_1) \rightarrow (o_1, \tau, o)$
rdfs5	$(p_1, \prec_{sp}, p_2), (p_2, \prec_{sp}, p_3) \rightarrow (p_1, \prec_{sp}, p_3)$
rdfs7	$(p_1, \prec_{sp}, p_2), (b_s, p_1, o) \rightarrow (b_s, p_2, o)$
rdfs9	$(b_s, \prec_{sc}, o), (b_{s_1}, \tau, b_s) \rightarrow (b_{s_1}, \tau, o)$
rdfs11	$(b_s, \prec_{sc}, o), (o, \prec_{sc}, o_1) \rightarrow (b_s, \prec_{sc}, o_1)$
ext1	$(p, \hookleftarrow_d, o), (o, \prec_{sc}, o_1) \rightarrow (p, \hookleftarrow_d, o_1)$
ext2	$(p, \hookrightarrow_r, o), (o, \prec_{sc}, o_1) \rightarrow (p, \hookrightarrow_r, o_1)$
ext3	$(p, \prec_{sp}, p_1), (p_1, \hookleftarrow_d, o) \rightarrow (p, \hookleftarrow_d, o)$
ext4	$(p, \prec_{sp}, p_1), (p_1, \hookrightarrow_r, o) \rightarrow (p, \hookrightarrow_r, o)$

Table 1: RDF & RDFS statements (left) and sample RDF entailment rules (right).

abstract

1 Introduction

To facilitate working with very large, complex-structure, heterogeneous graphs, many **graph summaries** have been proposed, including some specifically tailored for Resource Description Framework (RDF) graphs [1, 4, 5]. A summary of an RDF graph G is a smaller graph (typically also RDF), based on which questions about G may be answered more efficiently than by using G directly.

In this work, we define a *formal generic summarization framework* for RDF graphs, based on the classical notion of graph quotients, and on our notion of *RDF node equivalence*. While quotient-style summaries have been studied in the past [1, 2], our first contribution is a formal framework for summarizing RDF graphs including possible *RDF Schema constraints*, which leads us to study *the interplay between summarization and saturation* with such constraints. Specifically, our second contribution is *a sufficient condition* on the RDF node equivalence relation, which guarantees that the summary of the saturation of G can be built through a *shortcut* procedure, without actually saturating G; this can significantly speed up the summary construction.

Our summaries, representative of the complete (saturated) graphs but often much smaller, can be used in GUIs to help users explore and query RDF graphs, or to optimize structured and/or keyword queries etc. as has been done in previous works [2, 3, 4, 5].

2 Preliminaries

An RDF graph is a set of triples (s, p, o) where *s* is termed the *subject*, *p* the *property*, and *o* the *object*; such a triple states that s is described with the property p that has value o. *Well-formed* triples, as per the RDF specification, belong to $(\mathcal{U} \cup \mathcal{B}) \times \mathcal{U} \times (\mathcal{U} \cup \mathcal{L} \cup \mathcal{B})$, where $\mathcal{U}$ is a set of *Universal Resource Identifiers* (URIs in short), $\mathcal{L}$ is a set of *literals* (constants), and $\mathcal{B}$ is a set of *blank nodes*, representing unknown URI or literal values. A triple (s, p, o) states that its subject s has the property p whose value is the object o. RDF allows making *class assertions*, if p is the special built-in RDF property rdf:type (τ in short), and *property assertions* otherwise (Table 1).

RDF Schema statements (at the bottom left of Table 1, together with the shorthand notations of their properties) allow specifying ontological constraints relating classes and/or properties. The semantics of an RDF graph G is its *saturation* (or *closure*) G^∞, defined as the G triples together with all the *implicit* triples that can be derived from them and the entailment rules from the RDF standard. Table 1 (right) shows rules that use RDFS constraints to derive implicit facts or implicit constraints. Figure 1 depicts a sample publications graph, where Pub stands for publication ($CPub$ in conferences and $JPub$ in journals), a for

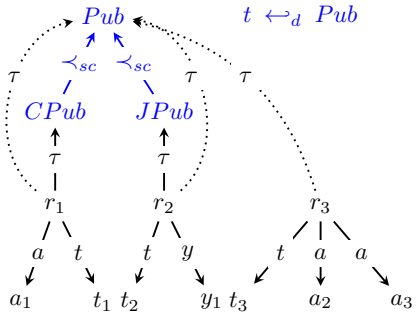


Figure 1: Sample RDF graph G.

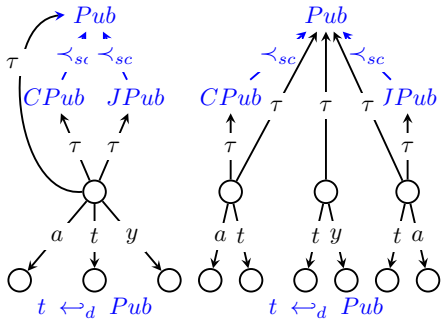


Figure 2: Sample summaries.

author, t for title and y for year; class nodes and RDFS triples appear in blue, for instance, the domain of t (title) is Pub. Solid arrows correspond to explicit G triples, and dotted arrows to *implicit* triples; all together, they depict $\mathtt{G}^{\infty}$.

3 Summarization framework

We start by recalling the classical notion of graph quotient. Let $G = (V, E)$ be a labeled directed graph whose edges E have labels from a set A. Let $\sim$ be an equivalence relation over the graph node set V. The *quotient of G through $\sim$*, denoted $G_{/\sim}$, is a labeled directed graph having (i) a node n_S for each set S of equivalent V nodes, and (ii) an edge $n_{S_1} \xrightarrow{a} n_{S_2}$ for some label $a \in A$ iff there exist two V nodes $n_1 \in S_1$ and $n_2 \in S_2$ such that the edge $n_1 \xrightarrow{a} n_2 \in E$.

When summarizing an RDF graph, class and schema information (e.g., the blue part of Figure 1) should be preserved, as they encode its semantics. Thus, we define:

Definition 1. (RDF EQUIVALENCE RELATION) *Let $\equiv$ be a binary relation between the nodes of an RDF graph. We say $\equiv$ is an* RDF equivalence relation *iff (i) $\equiv$ is reflexive, symmetric and transitive, (ii) any class node is $\equiv$ only to itself, and (iii) any property node is $\equiv$ only to itself.*

We define an RDF summary as a graph quotient w.r.t. a given RDF node equivalence:

Definition 2. (RDF SUMMARY) *Given an RDF graph G and an RDF node equivalence relation $\equiv$, the* summary of G by $\equiv$*, which is an RDF graph denoted $\mathtt{G}_{/\equiv}$, is the quotient of G by $\equiv$. $\mathtt{G}_{/\equiv}$ data nodes use fresh URIs, one for each set of equivalent G data nodes.*

Different RDF equivalence relations lead to different summaries. Figures 2 illustrates two possible summaries, on the saturated $\mathtt{G}^\infty$ from Figure 1; circles denote new-URI summary nodes, each of which represents a set of G nodes. For instance, at left, a single node represents r_1, r_2, r_3; at right, they are separated by their sets of types. Below, we do not discuss any particular summary further; instead, we focus on our summarization framework, and its interplay with saturation.

For a summary to reflect (*represent*) a graph G, queries having answers on G should also have answers on the summary. Given an *RDF query language* $\mathcal{Q}$, we define:

Definition 3. (SUMMARY REPRESENTATIVENESS) *Let* G *be any RDF graph.* $\mathtt{G}_{/\equiv}$ *is* $\mathcal{Q}$-representative of G *if and only if for any query* $q \in \mathcal{Q}$ *such that* $q(\mathtt{G}^\infty) \neq \emptyset$*, we have* $q((\mathtt{G}_{/\equiv})^\infty) \neq \emptyset$*.*

We target summaries representative of any query over the *graph structure* of G, including imprecise queries using variables in some property positions. Thus, we instantiate $\mathcal{Q}$ into Extended Relational Basic Graph Pattern Queries (*RBGP**, in short), a core fragment of SPARQL, defined as follows. A *query triple pattern* is an element of $\mathcal{V} \times (\mathcal{U} \cup \mathcal{V}) \times \mathcal{V}$, where $\mathcal{V}$ is a set of variables. An RBGP* query q is of the form: $q(\bar{x}) \leftarrow t_1, \ldots, t_n$ where each t_i is a query triple pattern, $\{t_1, \ldots, t_n\}$ is noted $body(q)$, and $\bar{x}$, called the *answer variables*, is a subset of the variables in $body(q)$. A sample RBGP* query is: $q^*(x_1, x_3)$:- $(x_1, \tau, \mathtt{Book}),\ (x_1, \mathtt{author}, x_2),\ x_2\ y\ x_3$.

We show (the proof appears in the Appendix, Section 5.1) that for any RDF equivalence relation $\equiv$:

Proposition 1. (SUMMARY REPRESENTATIVENESS) *An RDF summary* $\mathtt{G}_{/\equiv}$ *is RBGP*-representative.*

RBGP* representativeness ensures that any query specifying a certain graph pattern in G and/or querying the structure itself (by means of variables in property positions, such as y in the sample query above) which has answers on G, also has answers on $\mathtt{G}_{/\equiv}$.

In the presence of an RDF Schema, the semantics of G is its saturation $\mathtt{G}^\infty$. Thus, a representative summary must reflect both the explicit and the implicit triples of G. For instance, the summaries in Figure 2 show that some $\mathtt{G}^\infty$ resources (e.g., r_1, r_2, r_3) are of type Pub, but the same summaries computed from G alone do not, as the corresponding τ triples are implicit in G. A simple way to obtain $(\mathtt{G}^\infty)_{/\equiv}$ is to compute $\mathtt{G}^\infty$ and then summarize it. We define a novel alternative *shortcut* method, which avoids saturating G, yet it constructs an RDF graph *strongly isomorphic* to $(\mathtt{G}^\infty)_{/\equiv}$, as follows:

Definition 4. (STRONG ISOMORPHISM) *A* strong isomorphism between two RDF graphs $\mathtt{G}_1, \mathtt{G}_2$*, noted* $\mathtt{G}_1 \simeq \mathtt{G}_2$*, is an isomorphism which is the identity for the class and property nodes.*

Definition 5. (SUMMARY COMPLETENESS) *Summarization through the RDF node equivalence relation* $\equiv$ admits a shortcut *iff for any RDF graph* G*,* $(\mathtt{G}^\infty)_{/\equiv} \simeq ((\mathtt{G}_{/\equiv})^\infty)_{/\equiv}$ *holds.*

The shortcut summarizes G, saturates the result, then summarize it again (the three green edges in Figure 3). Its result is *essentially* $(\mathtt{G}^\infty)_{/\equiv}$, as the two have identical graph structures (guaranteed by the strong isomorphism), on which RBGP* representativeness is defined. They only differ in the new URIs of their nodes (circles in Figure 2).

What is the interest of the shortcut? If $\mathtt{G}_{/\equiv}$ is much smaller than G, it is much faster to saturate $\mathtt{G}_{/\equiv}$ (on the shortcut) than to saturate G; $(\mathtt{G}_{/\equiv})^\infty$ is also likely to be small, thus fast to summarize. Further, summarizing $\mathtt{G}^\infty$ is faster than summarizing G, given that $\mathtt{G}^\infty$ is at least as large as G. Summing up these inequalities, *the time spent on the shortcut may be (much) shorter than the time spent to build* $(\mathtt{G}^\infty)_{/\equiv}$ *directly*.

By the summary definition, to every node in G corresponds exactly one node in the summary $\mathtt{G}_{\equiv}$. We call **representation function** and denote $f_\equiv$ the function associating a summary node to each G node; we say $f_\equiv(n)$ *represents* n in the summary. An important structural property relates G, $\mathtt{G}^\infty$ and the function $f_\equiv$ (see Figure 3):

Lemma 1 (Summarization Homomorphism)**.** *Let* G *be an RDF graph,* $\mathtt{G}_{/\equiv}$ *its summary and* $f_\equiv$ *the corresponding representation function from* G *nodes to* $\mathtt{G}_{/\equiv}$ *nodes.* $f_\equiv$ *defines a homomorphism from* $\mathtt{G}^\infty$ *to* $(\mathtt{G}_{/\equiv})^\infty$*.*

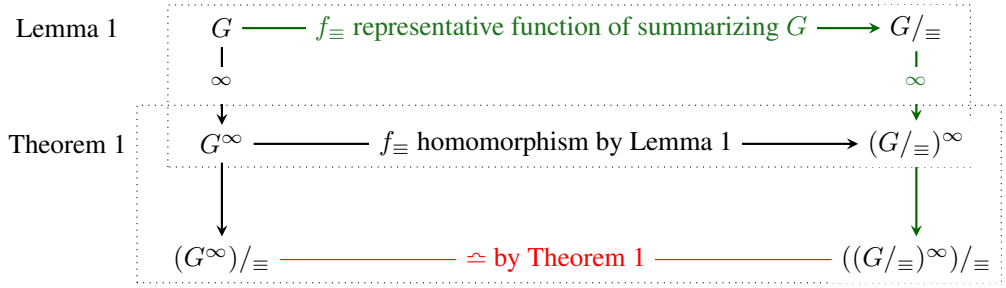


Figure 3: Illustration for Lemma 1 and Theorem 1.

Based on the Lemma, we establish the sufficient condition (see Figure 3):

Theorem 1 (Sufficient condition for shortcuts)**.** *Given an RDF node equivalence relation $\equiv$, and an RDF graph $\mathtt{G}$, let $\mathtt{G}_{/\equiv}$ be its summary and $f_{\equiv}$ the corresponding representation function from $\mathtt{G}$ nodes to $\mathtt{G}_{/\equiv}$ nodes.*

If $\equiv$ satisfies: for any RDF graph $\mathtt{G}$ and any pair (n_1, n_2) of $\mathtt{G}$ nodes, $n_1 \equiv n_2$ in $\mathtt{G}^{\infty}$ iff $f_{\equiv}(n_1) \equiv f_{\equiv}(n_2)$ in $(\mathtt{G}_{/\equiv})^{\infty}$, then $(\mathtt{G}^{\infty})_{/\equiv} \backsimeq ((\mathtt{G}_{/\equiv})^{\infty})_{/\equiv}$ holds.

The proof appears in the Appendix (Section 5.2).

The summary illustrated at left in Figure 2 turns out to admit the shortcut; in our experiments, the shortcut was up to **20x faster** than saturating $\mathtt{G}$ and then summarizing $\mathtt{G}^{\infty}$. The summary illustrated at right in Figure 2 does not admit the shortcut.

4 Conclusion and perspectives

Finding a necessary (and sufficient) condition for the shortcut is currently open. We have instantiated our framework into many summaries, and implemented a summarization tool available for download (together with many sample summaries) at https://team.inria.fr/cedar/projects/rdfsummary. We are currently working on summary-based query pruning, where we decide based on $(\mathtt{G}^{\infty})_{/\equiv}$ whether a query may have answers on $\mathtt{G}^{\infty}$ or not.

References

[1] S. Campinas, R. Delbru, and G. Tummarello. Efficiency and precision trade-offs in graph summary algorithms. In *IDEAS*, 2013.

[2] Q. Chen, A. Lim, and K. W. Ong. $D(K)$-index: An adaptive structural summary for graph-structured data. In *SIGMOD*, 2003.

[3] S. Gurajada, S. Seufert, I. Miliaraki, and M. Theobald. Using graph summarization for join-ahead pruning in a distributed RDF engine. In *SWIM*, 2014.

[4] T. Tran, G. Ladwig, and S. Rudolph. Managing structured and semistructured RDF data using structure indexes. *IEEE TKDE*, 25(9), 2013.

[5] G. Troullinou, H. Kondylakis, E. Daskalaki, and D. Plexousakis. RDF digest: Efficient summarization of RDF/S KBs. In *ESWC*, 2015.

5 Appendix

5.1 Proof of Proposition 1

For the purpose of the proof, we introduce a simple class of RDF queries, namely RBGPs (below):

Definition 6. (RELATIONAL BGP (RBGP) QUERIES) *A* relational BGP *(RBGP, in short) query is a BGP query whose body has: (i) URIs in all the property positions, (ii) a URI in the object position of every τ triple, and (iii) variables in any other positions.*

Proof. We prove the statement for **RBGP** queries; Proposition 2 (below) carries the statement over to RBGP*.

Let q be a query such that $q(\mathtt{G}^\infty) \neq \emptyset$; we need to show that $q((\mathtt{G}_{/\equiv})^\infty) \neq \emptyset$.

Let $\phi : q \to \mathtt{G}^\infty$ be an embedding, assigning to each query variable v, a node from $\mathtt{G}^\infty$; we extend ϕ to say it maps triple patterns from q into triples from $\mathtt{G}^\infty$. We need to produce an embedding from q into $(\mathtt{G}_{/\equiv})^\infty$.

First, consider a triple pattern t of q whose embedding $\phi(t) \in \mathtt{G}^\infty$ also belongs to $\mathtt{G}$.

- If $\phi(t)$ is a schema triple, then $\phi(t)$ is also in $\mathtt{G}_{/\equiv}$, since $\mathtt{G}$ and $\mathtt{G}_{/\equiv}$ have the same schema triples.
- Else if $\phi(t)$ is a type triple of the form $\mathtt{s}\ \tau\ \mathtt{c}$, the triple $f_\equiv(\mathtt{s})\ \tau\ \mathtt{c}$ belongs to $\mathtt{G}_{/\equiv}$, thus also to $(\mathtt{G}_{/\equiv})^\infty$.
- Otherwise, $\phi(t) = \mathtt{s\ p\ o} \in \mathtt{G}$ is a data triple, $\mathtt{G}_{/\equiv}$ holds the triple $f_\equiv(\mathtt{s})\ \mathtt{p}\ f_\equiv(\mathtt{o})$, thus $(\mathtt{G}_{/\equiv})^\infty$ also comprises it.

Now consider a triple pattern t' of q whose embedding $\phi(t') \in \mathtt{G}^\infty$ does not belong to $\mathtt{G}$.

- If $\phi(t')$ is a *schema* triple, then by definition of RDF entailment $\phi(t') \in (\mathtt{S_G})^\infty$, thus it is also in $(\mathtt{G}_{/\equiv})^\infty$ since $\mathtt{S_G} = \mathtt{S}_{\mathtt{G}_{/\equiv}} \subseteq \mathtt{G}_{/\equiv}$ by definition of a summary.
- Else if $\phi(t')$ is a *data* triple in $\mathtt{G}^\infty$, then by definition of RDF entailment, this triple $\phi(t')$ must be entailed by a *data* triple $t_d = \mathtt{s}_d\ \mathtt{p}_d\ \mathtt{o}_d$ in $\mathtt{G}$ and a subproperty constraints t_s, i.e., a schema triple in $\mathtt{S}^\infty$. As explained above, $f_\equiv(\mathtt{s}_d)\ \mathtt{p}_d\ f_\equiv(\mathtt{o}_d) \in \mathtt{G}_{/\equiv}$; at the same time, t_s also belongs $(\mathtt{G}_{/\equiv})^\infty$, since $\mathtt{S_G} = \mathtt{S}_{\mathtt{G}_{/\equiv}}$ hence $(\mathtt{S_G})^\infty = (\mathtt{S}_{\mathtt{G}_{/\equiv}})^\infty$. It follows that the inference step which entailed $\phi(t')$ from t_d and t_s in $\mathtt{G}^\infty$ also applies on $f_\equiv(\mathtt{s})\ \mathtt{p}_d\ f_\equiv(\mathtt{o}_d)$ and t_s in $(\mathtt{G}_{/\equiv})^\infty$.
- Otherwise, $\phi(t')$ is a τ triple in $\mathtt{G}^\infty$. This may result either:
  - from a $\mathtt{D_G}$ triple $t_d = \mathtt{s}_d\ \mathtt{p}_d\ \mathtt{o}_d$ and a triple $t_s \in (\mathtt{S_G})^\infty$, if t_s is a $\hookleftarrow_d$ or $\hookrightarrow_r$ triple. This case is very similar to the one above.
  - from a $\mathtt{T_G}$ triple of the form $\mathtt{s}\ \tau\ \mathtt{c}_1$ and a schema triple $t_s = \mathtt{c}_1 \prec_{sc} \mathtt{c}_2 \in (\mathtt{S_G})^\infty$, such that $\phi(t') = \mathtt{s}\ \tau\ \mathtt{c}_2$. In this case, $\mathtt{G}_{/\equiv}$ holds the triple $f_\equiv(\mathtt{s})\ \tau\ \mathtt{c}_1$ which is also present in $(\mathtt{G}_{/\equiv})^\infty$, thus the same inference step applies in $(\mathtt{G}_{/\equiv})^\infty$ to produce $f_\equiv(\mathtt{s})\ \tau\ \mathtt{c}_2$, since $\mathtt{S_G} = \mathtt{S}_{\mathtt{G}_{/\equiv}}$ hence $(\mathtt{S_G})^\infty = (\mathtt{S}_{\mathtt{G}_{/\equiv}})^\infty$.

Thus, any q triple mapped by ϕ into a data $\mathtt{G}^\infty$ triple (which may or may not explicitly belong to $\mathtt{G}$) is also mapped into a corresponding triple in $(\mathtt{G}_{/\equiv})^\infty$.

To conclude this proof, we now need to show that, in addition to the fact that each q triple that has an embedding in $\mathtt{G}^\infty$ has also necessarily an embedding in $\mathtt{G}^\infty_{/\equiv}$, if q has an embedding in $\mathtt{G}^\infty$, then q has also an embedding in $(\mathtt{G}_{/\equiv})^\infty$. This amounts to show that any two q triples t_1 and t_2 that join and that have an embedding in $\mathtt{G}^\infty$ also embed in $(\mathtt{G}_{/\equiv})^\infty$ (i.e., q joins are preserved).

- If both $\phi(t_1)$ and $\phi(t_2)$ are schema triples in $\mathtt{G}^\infty$, then these two triples are also in $(\mathtt{G}_{/\equiv})^\infty$, since $\mathtt{S_G} = \mathtt{S}_{\mathtt{G}_{/\equiv}}$ hence $(\mathtt{S_G})^\infty = \mathtt{S}^\infty_{\mathtt{G}_{/\equiv}}$.

- Else if both $\phi(t_1)$ and $\phi(t_2)$ are non-schema triples in $\mathtt{G}^\infty$:

  - If both $\phi(t_1)$ and $\phi(t_2)$ are data triples in $\mathtt{G}^\infty$, there exists a triple t'_1) (respt'_2) in $\mathtt{G}$, with same subject/object, from which $\phi(t_1)$ (resp. $\phi(t_2)$) is entailed using a sub-property constraint t^1_s (resp. t^2_s) from $\mathtt{S}^\infty$. Since $\phi(t_1)$ and t'_1 (resp. $\phi(t_2)$ and t'_2) have the same subject and object values, then t'_1 and t'_2 have same values on the places where t_1 and t_2 join. Therefore, if we assume that $t'_1 = \mathtt{s}_1\ \mathtt{p}_1\ \mathtt{o}_1$ and $t'_2 = \mathtt{s}_2\ \mathtt{p}_2\ \mathtt{o}_2$, the $\mathtt{G}_{/\equiv}$ triples $f_\equiv(\mathtt{s}_1)\ \mathtt{p}_1\ f_\equiv(\mathtt{o}_1)$ and $f_\equiv(\mathtt{s}_2)\ \mathtt{p}_2\ f_\equiv(\mathtt{o}_2)$ necessarily have same values on the places where t_1 and t_2 join. Moreover, since $\mathtt{S}_\mathtt{G} = \mathtt{S}_{\mathtt{G}_{/\equiv}}$ hence $(\mathtt{S}_\mathtt{G})^\infty = (\mathtt{S}_{\mathtt{G}_{/\equiv}})^\infty$, these two $\mathtt{G}_{/\equiv}$ triples and the above-mentioned t^1_s and t^2_s schema triples, produce the counterpart triples of $\phi(t_1)$ and $\phi(t_2)$ in $(\mathtt{G}_{/\equiv})^\infty$, which have same subject and object values. Thus, the q triples t_1 and t_2 embed in these two $(\mathtt{G}_{/\equiv})^\infty$ triples, if they embed in $\phi(t_1)$ and $\phi(t_2)$ in $\mathtt{G}^\infty$.

  - Else if both $\phi(t_1)$ and $\phi(t_2)$ are type triples in $\mathtt{G}^\infty$, say $\phi(t_1) = u\ \tau\ c_1$ and $\phi(t_2) = u\ \tau\ c_2$, then $t_1 = x\ \tau\ c_1$ and $t_2 = x\ \tau\ c_2$ by definition of an RBGP query. As in the cases of single q triple embeddings, $\phi(t_1)$ (resp. $\phi(t_2)$) results either from a $\mathtt{G}$ triple $u\ \tau\ c$ and a $\mathtt{S}^\infty_\mathtt{G}$ triple $c\ \prec_{sc}\ c_1$, or a $\mathtt{G}$ triple $u\ p\ u_1$ and a $\mathtt{S}^\infty_\mathtt{G}$ triple $p\ \hookleftarrow_d\ c_1$, or a $\mathtt{G}$ triple $u_1\ p\ u$ and a $(\mathtt{S}_\mathtt{G})^\infty$ triple $p\ \hookrightarrow_r\ c_1$. Therefore, since $\mathtt{S}_\mathtt{G} = \mathtt{S}_{\mathtt{G}_{/\equiv}}$ hence $\mathtt{S}^\infty_\mathtt{G} = (\mathtt{S}_{\mathtt{G}_{/\equiv}})^\infty$, for $\phi(t_1)$ (resp. $\phi(t_2)$), there are either a $\mathtt{G}_{/\equiv}$ triple $f_\equiv(u)\ \tau\ c$ and a $(\mathtt{S}_{\mathtt{G}_{/\equiv}})^\infty$ triple $c\ \prec_{sc}\ c_1$, or a $\mathtt{G}_{/\equiv}$ triple $f_\equiv(u)\ p\ f_\equiv(u_1)$ and a $(\mathtt{S}_{\mathtt{G}_{/\equiv}})^\infty$ triple $p\ \hookleftarrow_d\ c_1$, or a $\mathtt{G}_{/\equiv}$ triple $f_\equiv(u_1)\ p\ f_\equiv(u)$ and a $(\mathtt{S}_{\mathtt{G}_{/\equiv}})^\infty$ triple $p\ \hookrightarrow_r\ c_1$, which entail $f_\equiv(u)\ \tau\ c_1$ and $f_\equiv(u)\ \tau\ c_2$ in $\mathtt{G}^\infty_{/\equiv}$. Thus, the q triples t_1 and t_2 embed in these two $\mathtt{G}^\infty_{/\equiv}$ triples, if they embed in $\phi(t_1)$ and $\phi(t_2)$ in $\mathtt{G}^\infty$.

  - Otherwise, $\phi(t_1)$ is a data triple in $\mathtt{G}^\infty$ and $\phi(t_2)$ is a type triple in $\mathtt{G}^\infty$. This case is very similar to the two above case, hence we do not detail it.

- Otherwise, $\phi(t_1)$ is a schema triple in $\mathtt{G}^\infty$ and $\phi(t_2)$ is not a schema triples in $\mathtt{G}^\infty$. In this case, since q is an RBGP query, q triples must be such that t_1 is $\mathtt{s}_1\ \mathtt{p}_1\ \mathtt{o}_1$ with $\mathtt{p}_1 \in \{\prec_{sc}, \prec_{sp}, \hookleftarrow_d, \hookrightarrow_r\}$ and t_2 is either $\mathtt{s}_2\ p\ \mathtt{o}_2$ or $\mathtt{s}_2\ \tau\ c$.
  Since $\mathtt{G}^\infty$ and $\mathtt{G}^\infty_{/\equiv}$ have the same schema, $\phi(t_1)$ also belongs to $\mathtt{G}^\infty_{/\equiv}$. Now:

  - If t_2 is $\mathtt{s}_2\ p\ \mathtt{o}_2$ then $\phi(t_2) = s, p, o$ must be either in $\mathtt{G}$ or entailed from a $\mathtt{G}$ data triple s, p', o and a $\mathtt{S}^\infty_\mathtt{G}$ sub-property triple $p', \prec_{sp}, p$ (see above, for single q triple embedding). If $\phi(t_2)$ is in $\mathtt{G}$, then $f_\equiv(s)\ p\ f_\equiv(o)$ is in $\mathtt{G}_{/\equiv}$, hence in $\mathtt{G}^\infty_{/\equiv}$. Otherwise, $f_\equiv(s)\ p'\ f_\equiv(o)$ is in $\mathtt{G}_{/\equiv}$, $p'\ \prec_{sp}\ p$ is in $\mathtt{S}_{\mathtt{G}_{/\equiv}}$ (since $\mathtt{G}$ and $\mathtt{G}_{/\equiv}$ have the same schema), thus $f_\equiv(s)\ p\ f_\equiv(o)$ is in $\mathtt{G}^\infty_{/\equiv}$. Since t_1 and t_2 joins, s and/or p are class/property nodes. If s (resp. o) is a class/property node, then $f_\equiv(s) = s$ (resp. $f_\equiv(o) = o$). Hence, $\phi(t_1) \in \mathtt{G}^\infty_{/\equiv}$ joins with $f_\equiv(s)\ p\ f_\equiv(o) \in \mathtt{G}^\infty_{/\equiv}$.

  - If t_2 is $\mathtt{s}_2\ \tau\ c$ then $\phi(t_2) = s\ \tau\ c$ must be either in $\mathtt{G}$ or entailed from (i) a $\mathtt{G}$ data triple $s\ p\ o$ and a $\mathtt{S}^\infty_\mathtt{G}$ triple $p\ \hookleftarrow_d\ c$, or a $\mathtt{G}$ data triple $s_1\ p\ s$ and a $\mathtt{S}^\infty_\mathtt{G}$ triple $p\ \hookrightarrow_r\ c$, or (iii) a $\mathtt{G}$ type triple $s\ \tau\ c'$ and a $\mathtt{S}^\infty_\mathtt{G}$ triple $c'\ \prec_{sc}\ c$ (see above, for single q triple embedding). If $\phi(t_2)$ is in $\mathtt{G}$, then $f_\equiv(s)\ \tau\ c$ is in $\mathtt{G}_{/\equiv}$, hence in $\mathtt{G}^\infty_{/\equiv}$. Otherwise, $f_\equiv(s)\ p\ f_\equiv(o)$ or $f_\equiv(s_1)\ p\ f_\equiv(s)$ or $f_\equiv(s)\ \tau\ c'$ is in $\mathtt{G}_{/\equiv}$, and (since $\mathtt{G}$ and $\mathtt{G}_{/\equiv}$ have the same schema) thus $f_\equiv(s)\ \tau\ c$ is in $\mathtt{G}^\infty_{/\equiv}$. Since t_1 and t_2 can only join on $\mathtt{s}_2$, s is a class or property nodes, hence $f_\equiv(s) = s$. Therefore, $\phi(t_1) \in \mathtt{G}^\infty_{/\equiv}$ joins with $f_\equiv(s)\ \tau\ c \in \mathtt{G}^\infty_{/\equiv}$.

□

Proposition 2. (RBGP VS. RBGP* REPRESENTATIVENESS) *RBGP representativeness entails RBGP* representativeness.*

Proof. Let q^* be an RBGP* query which is non-empty on $\mathtt{G}^\infty$ and $\mathtt{G}_{/\equiv}$ be an RBGP-representative summary of $\mathtt{G}$. We show that non-emptiness of q^* on $\mathtt{G}^\infty$ entails its non-emptiness on $(\mathtt{G}_{/\equiv})^\infty$.
Given that $q^*(\mathtt{G}^\infty)$ is non-empty, there exists at least an embedding of q^* into $\mathtt{G}^\infty$; let q be the query obtained by replacing in q^*, each variable occurring in the property position by the concrete property matching it in $\mathtt{G}^\infty$. Clearly, q has results on $\mathtt{G}^\infty$, and since $\mathtt{G}_{/\equiv}$ is RBGP representative, q also has results on $(\mathtt{G}_{/\equiv})^\infty$. Therefore, $q(\mathtt{G}^\infty_{/\equiv}) \neq \emptyset$, and given that $q \subseteq q^*$ (query containment), it follows that $q^*((\mathtt{G}_{/\equiv})^\infty) \neq \emptyset$. □

5.2 Proof of Lemma 1

We first establish:

Proposition 3. (CLASS, PROPERTY AND SCHEMA PRESERVATION) *An RDF graph* $\mathtt{G}$ *and an RDF summary* $\mathtt{G}_{/\equiv}$ *of it have the same sets of classes names, of property names, and of schema triples.*

Indeed, Definitions 1 and 2 ensure that class and property nodes are preserved through summarization, as well as their URI labels, since they cannot be equivalent (hence fused) with other nodes. Further, because our summarization approach relies on graph quotients, all property names labelling edges in a given graph also label edges in its summary. This obviously implies that schema triples are preserved, as they only involve class or property nodes.

The Lemma is proved as follows (recall also Figure 3):

Proof. We first show that an homomorphism can be established from the node sets of $\mathtt{G}^\infty$ to that of $(\mathtt{G}_{/\equiv})^\infty$.

Observe that RDF saturation with RDFS constraints only adds edges between graph nodes, but does not add nodes. Thus, a node n is in $\mathtt{G}^\infty$ iff n is in $\mathtt{G}$. Further, by the definition of our quotient-based summaries (Definition 2), n is in $\mathtt{G}$ iff $f_\equiv(n)$ is in $\mathtt{G}_{/\equiv}$. Finally, again by the definition of saturation, $f_\equiv(n)$ is in $\mathtt{G}_{/\equiv}$ iff $f_\equiv(n)$ is in $(\mathtt{G}_{/\equiv})^\infty$.

Therefore, every $\mathtt{G}^\infty$ node n maps the $f_\equiv(n)$ $(\mathtt{G}_{/\equiv})^\infty$ node (*).

Next, we show that there is a one-to-one mapping between $\mathtt{G}^\infty$ edges and those of $(\mathtt{G}_{/\equiv})^\infty$.

If $n_1\ p\ n_2$ is an edge in $\mathtt{G}^\infty$, at least one of the following two situations holds:

- $n_1\ p\ n_2$ is an edge in $\mathtt{G}$. This holds iff $f_\equiv(n_1)\ p\ f_\equiv(n_2)$ is an edge in $\mathtt{G}_{/\equiv}$, by definition of an RDF summary. Finally, if $f_\equiv(n_1)\ p\ f_\equiv(n_2)$ is an edge in $\mathtt{G}_{/\equiv}$, then $f_\equiv(n_1)\ p\ f_\equiv(n_2)$ is also an edge in $(\mathtt{G}_{/\equiv})^\infty$.
- $n_1\ p'\ n_2$ is an edge in $\mathtt{G}$, and $p' \prec_{sp} p$ is in $\mathtt{S}^\infty_\mathtt{G}$, thus $n_1\ p\ n_2$ is produced by saturation in $\mathtt{G}^\infty$. In this case, we show similarly to the preceding item that $f_\equiv(n_1)\ p'\ f_\equiv(n_2)$ is an edge in $(\mathtt{G}_{/\equiv})^\infty$, hence $f_\equiv(n_1)\ p\ f_\equiv(n_2)$ is also an edge added to $(\mathtt{G}_{/\equiv})^\infty$ by saturation, since $(\mathtt{G}_{/\equiv})^\infty$ and $\mathtt{G}^\infty$ have the same (saturated) schema triples (Property 3).

If $n_1\ \tau\ c$ is an edge in $\mathtt{G}^\infty$, at least one of the following two situations holds:

- $n_1\ \tau\ c$ is an edge in $\mathtt{G}$. This holds iff $f_\equiv(n_1)\ \tau\ c$ is an edge in $\mathtt{G}_{/\equiv}$, by definition of an RDF summary (recall that $f_\equiv(c) = c$ for classes). Finally, if $f_\equiv(n_1)\ \tau\ c$ is an edge in $\mathtt{G}_{/\equiv}$, then $f_\equiv(n_1)\ \tau\ c$ is also an edge in $(\mathtt{G}_{/\equiv})^\infty$.
- $n_1\ p\ n_2$ is an edge in $\mathtt{G}$ and $p \hookleftarrow_d c$ (or $p \hookrightarrow_r c$) is in $\mathtt{S}^\infty_\mathtt{G}$, thus $n_1\ \tau\ c$ is produced by saturation in $\mathtt{G}^\infty$. In this case, we show similarly as above that $f_\equiv(n_1)\ p\ f_\equiv(n_2)$ is an edge in $(\mathtt{G}_{/\equiv})^\infty$, hence $f_\equiv(n_1)\ \tau\ c$ is also an edge added to $(\mathtt{G}_{/\equiv})^\infty$ by saturation, since $(\mathtt{G}_{/\equiv})^\infty$ and $\mathtt{G}^\infty$ have the same (saturated) schema triples (Property 3).

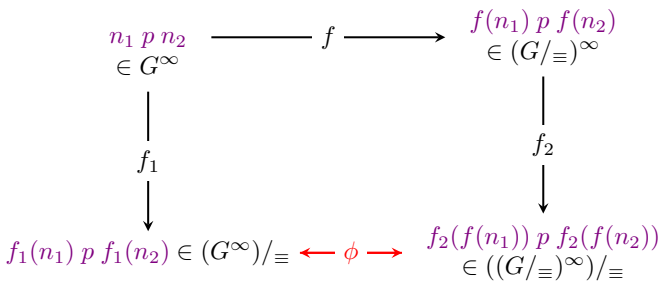


Figure 4: Diagram for the proof of Theorem 1.

Therefore, every $\mathtt{G}^\infty$ edge $n_1\ p\ n_2$ (resp. $n_1\ \tau\ c$) maps into the $(\mathtt{G}_{/\equiv})^\infty$ edge $f_\equiv(n_1)\ p\ f_\equiv(n_2)$ (resp. $f_\equiv(n_1)\ \tau\ c$) (**).

From (*) and (**), it follows that f is an homomorphism from $\mathtt{G}^\infty$ to $(\mathtt{G}_{/\equiv})^\infty$. □

5.3 Proof of Theorem 1

Proof. We start by introducing some **notations** (see Figure 3). Let f_1 be the representation function from $\mathtt{G}^\infty$ into $(\mathtt{G}^\infty)_{/\equiv}$, and f_2 be the representation function from $(\mathtt{G}_{/\equiv})^\infty$ into $((\mathtt{G}_{/\equiv})^\infty)_{/\equiv}$.

Let the function φ be a function from the $(\mathtt{G}^\infty)_{/\equiv}$ nodes to the $((\mathtt{G}_{/\equiv})^\infty)_{/\equiv}$ nodes defined as: $\varphi(f_1(n)) = f_2(f(n))$ for n any $\mathtt{G}^\infty$ node.

Suppose that for every pair (n_1, n_2) of $\mathtt{G}$ nodes, $n_1 \equiv n_2$ in $\mathtt{G}^\infty$ iff $f(n_1) \equiv f(n_2)$ in $(\mathtt{G}_{/\equiv})^\infty$ holds. Let us show that this condition suffices to ensure $(\mathtt{G}^\infty)_{/\equiv} \equiv ((\mathtt{G}_{/\equiv})^\infty)_{/\equiv}$ holds, i.e., the φ function defines an isomorphism from $(\mathtt{G}^\infty)_{/\equiv}$ to $((\mathtt{G}_{/\equiv})^\infty)_{/\equiv}$.

First, let us show that φ is a bijection from all the $(\mathtt{G}^\infty)_{/\equiv}$ nodes to all the $((\mathtt{G}_{/\equiv})^\infty)_{/\equiv}$ nodes. Since for every pair n_1, n_2 of $\mathtt{G}^\infty$ nodes, $n_1 \equiv n_2$ iff $f(n_1) \equiv f(n_2)$ in $(\mathtt{G}_{/\equiv})^\infty$, it follows that $(\mathtt{G}^\infty)_{/\equiv}$ and $((\mathtt{G}_{/\equiv})^\infty)_{/\equiv}$ have the same number of nodes (*).

Further, a given node n in $(\mathtt{G}^\infty)_{/\equiv}$ represents a set of equivalent nodes $n_1, \ldots, n_k$ from $\mathtt{G}^\infty$. By hypothesis, $n_1 \equiv \cdots \equiv n_k$ in $\mathtt{G}^\infty$ iff $f(n_1) \equiv \cdots \equiv f(n_k)$ in $\mathtt{G}^\infty_{/\equiv}$ holds. Hence, every node $n = f_1(n_1) = \cdots = f_1(n_k)$ of $(\mathtt{G}^\infty)_{/\equiv}$ maps to a distinct node $n' = f_2(f(n_1)) = \cdots = f_2(f(n_k))$ in $((\mathtt{G}_{/\equiv})^\infty)_{/\equiv}$ (**).

Similarly, a given node n' in $((\mathtt{G}_{/\equiv})^\infty)_{/\equiv}$ represents a set of equivalent nodes $n'_1 = f(n_1), \ldots, n'_k = f(n_k)$ in $(\mathtt{G}_{/\equiv})^\infty$. By hypothesis, $f(n_1) \equiv \cdots \equiv f(n_k)$ in $\mathtt{G}^\infty_{/\equiv}$ iff $n_1 \equiv \cdots \equiv n_k$ in $\mathtt{G}^\infty$ holds. Hence, every node $n' = f_2(f(n_1)) = \cdots = f_2(f(n_k))$ in $((\mathtt{G}_{/\equiv})^\infty)_{/\equiv}$ maps to a distinct node $n = f_1(n_1) = \cdots = f_1(n_k)$ of $(\mathtt{G}^\infty)_{/\equiv}$ (***).

From (*), (**) and (***), it follows that φ is a bijective function from all the $(\mathtt{G}^\infty)_{/\equiv}$ nodes to all the $((\mathtt{G}_{/\equiv})^\infty)_{/\equiv}$ nodes.

Now, let us show that φ defines an isomorphism from $(\mathtt{G}^\infty)_{/\equiv}$ to $((\mathtt{G}_{/\equiv})^\infty)_{/\equiv}$.

For every edge $n'_1\ p\ n'_2$ in $(\mathtt{G}^\infty)_{/\equiv}$, by definition of an RDF summary, there exists an edge $n_1\ p\ n_2$ in $\mathtt{G}^\infty$ such that $n'_1\ p\ n'_2 = f_1(n_1)\ p\ f_1(n_2)$. Figure 4 illustrates the discussion. Further, if $n_1\ p\ n_2$ is in $\mathtt{G}^\infty$, then $f(n_1)\ p\ f(n_2)$ is in $(\mathtt{G}_{/\equiv})^\infty$ (Proposition 1), hence $f_2(f(n_1))\ p\ f_2(f(n_2))$ is in $((\mathtt{G}_{/\equiv})^\infty)_{/\equiv}$. Therefore,

- since for every $f_1(n_1)\ p\ f_1(n_2)$ edge in $(\mathtt{G}^\infty)_{/\equiv}$, there is an edge $f_2(f(n_1))\ p\ f_2(f(n_2))$ in $((\mathtt{G}_{/\equiv})^\infty)_{/\equiv}$, and
- since $\varphi(f_1(n)) = f_2(f(n))$, for n any $\mathtt{G}^\infty$ node, is a bijective function from all $(\mathtt{G}^\infty)_{/\equiv}$ nodes to all $((\mathtt{G}_{/\equiv})^\infty)_{/\equiv}$ nodes,

- it follows that $((\mathtt{G}_{/\equiv})^{\infty})_{/\equiv}$ contains the image of all $(\mathtt{G}^{\infty})_{/\equiv}$ $f_1(n_1)\ p\ f_1(n_2)$ triples through φ (*).

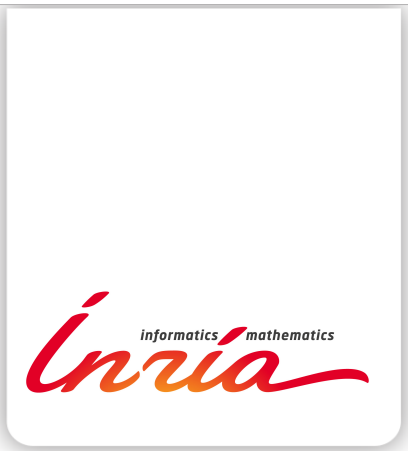
Now, for every edge $n_1''\ p\ n_2''$ in $((\mathtt{G}_{/\equiv})^{\infty})_{/\equiv}$, by definition of an RDF summary, there exists an edge $n_1'\ p\ n_2'$ in $(\mathtt{G}_{/\equiv})^{\infty}$ such that $n_1''\ p\ n_2'' = f_2(n_1')\ p\ f_2(n_2')$. Hence, by Proposition 1, there exists an edge $n_1\ p\ n_2$ in $\mathtt{G}^{\infty}$ such that $n_1'\ p\ n_2' = f(n_1)\ p\ f(n_2)$. Moreover, since $n_1\ p\ n_2$ is in $\mathtt{G}^{\infty}$, $f_1(n_1)\ p\ f_1(n_2)$ is in $(\mathtt{G}^{\infty})_{/\equiv}$. Therefore, since for every $f_2(f(n_1))\ p\ f_2(f(n_2))$ edge in $((\mathtt{G}_{/\equiv})^{\infty})_{/\equiv}$, there is an edge $f_1(n_1)\ p\ f_1(n_2)$ in $(\mathtt{G}^{\infty})_{/\equiv}$, and since $\varphi(f_1(n)) = f_2(f(n))$, for n any $\mathtt{G}^{\infty}$ node, is a bijective function from all $(\mathtt{G}^{\infty})_{/\equiv}$ nodes to all $((\mathtt{G}_{/\equiv})^{\infty})_{/\equiv}$ nodes, $(\mathtt{G}^{\infty})_{/\equiv}$ contains the image of all $((\mathtt{G}_{/\equiv})^{\infty})_{/\equiv}$ $n_1''\ p\ n_2''$ triples through φ^{-1} (**).

Similarly, for every edge $n_1'\ \tau\ c$ in $(\mathtt{G}^{\infty})_{/\equiv}$, by definition of an RDF summary, there exists an edge $n_1\ \tau\ c$ in $\mathtt{G}^{\infty}$ such that $n_1'\ \tau\ c = f_1(n_1)\ \tau\ c$. Further, if $n_1\ \tau\ c$ is in $\mathtt{G}^{\infty}$, then $f(n_1)\ \tau\ c$ is in $(\mathtt{G}_{/\equiv})^{\infty}$ (Proposition 1), hence $f_2(f(n_1))\ \tau\ c$ is in $((\mathtt{G}_{/\equiv})^{\infty})_{/\equiv}$. Therefore,

- since for every $f_1(n_1)\ \tau\ c$ edge in $(\mathtt{G}^{\infty})_{/\equiv}$, there is an edge $f_2(f(n_1))\ \tau\ c$ in $((\mathtt{G}_{/\equiv})^{\infty})_{/\equiv}$, and
- since $\varphi(f_1(n)) = f_2(f(n))$, for n any $\mathtt{G}^{\infty}$ node, is a bijective function from all $(\mathtt{G}^{\infty})_{/\equiv}$ nodes to all $((\mathtt{G}_{/\equiv})^{\infty})_{/\equiv}$ nodes,
- it follows that $((\mathtt{G}_{/\equiv})^{\infty})_{/\equiv}$ contains the image of all $(\mathtt{G}^{\infty})_{/\equiv}$ $f_1(n_1)\ \tau\ c$ triples through φ (*').

Now, for every edge $n_1''\ \tau\ c$ in $((\mathtt{G}_{/\equiv})^{\infty})_{/\equiv}$, by definition of an RDF summary, there exists an edge $n_1'\ \tau\ c$ in $(\mathtt{G}_{/\equiv})^{\infty}$ such that $n_1''\ \tau\ c = f_2(n_1')\ \tau\ c$. Hence, by Proposition 1, there exists an edge $n_1\ \tau\ c$ in $\mathtt{G}^{\infty}$ such that $n_1'\ \tau\ c = f(n_1)\ \tau\ c$. Moreover, since $n_1\ \tau\ c$ is in $\mathtt{G}^{\infty}$, $f_1(n_1)\ \tau\ c$ is in $(\mathtt{G}^{\infty})_{/\equiv}$. Therefore, since for every $f_2(f(n_1))\ \tau\ c$ edge in $((\mathtt{G}_{/\equiv})^{\infty})_{/\equiv}$, there is an edge $f_1(n_1)\ \tau\ c$ in $(\mathtt{G}^{\infty})_{/\equiv}$, and since $\varphi(f_1(n)) = f_2(f(n))$, for n any $\mathtt{G}^{\infty}$ node, is a bijective function from all $(\mathtt{G}^{\infty})_{/\equiv}$ nodes to all $((\mathtt{G}_{/\equiv})^{\infty})_{/\equiv}$ nodes, $(\mathtt{G}^{\infty})_{/\equiv}$ contains the image of all $((\mathtt{G}_{/\equiv})^{\infty})_{/\equiv}$ $n_1''\ \tau\ c$ triples through φ^{-1} (**').

From (*) and (**), and, (*') and (**'), it follows that φ defines an isomorphism from $(\mathtt{G}^{\infty})_{/\equiv}$ to $((\mathtt{G}_{/\equiv})^{\infty})_{/\equiv}$. □



RESEARCH CENTRE
SACLAY – ÎLE-DE-FRANCE

1 rue Honoré d'Estienne d'Orves
Bâtiment Alan Turing
Campus de l'École Polytechnique
91120 Palaiseau

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399