Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

Trust Models for RDF Data: Semantics and Complexity*

Valeria Fionda and Gianluigi Greco
Department of Mathematics and Computer Science,
University of Calabria, Italy
{fionda,ggreco}@mat.unical.it

Abstract

Due to the openness and decentralization of the Web, mechanisms to represent and reason about the reliability of RDF data become essential. This paper embarks on a formal analysis of RDF data enriched with trust information by focusing on the characterization of its model-theoretic semantics and on the study of relevant reasoning problems. The impact of trust values on the computational complexity of well-known concepts related to the entailment of RDF graphs is studied. In particular, islands of tractability are identified for classes of acyclic and nearly-acyclic graphs. Moreover, an implementation of the framework and an experimental evaluation on real data are discussed.

1 Introduction

The Resource Description Framework (RDF) is the basic representation language for the Semantic Web, where data is exposed as a set of triples expressing the properties that hold among a given universe of resources, identified by URIs (Hayes and Patel-Schneider 2014). In fact, RDF is not suited, in its basic form, to represent meta-information. Therefore, efforts have been made to define extensions for dealing with time (Gutierrez, Hurtado, and Vaisman 2007), provenance (Dividino et al. 2009), fuzzy (Straccia 2009), and trust (Hartig 2009; Tomaszuk, Pak, and Rybinski 2013). These ad-hoc extensions come in the form of annotations over the triples and general frameworks for annotated Semantic Web data have been proposed, too (Udrea, Recupero, and Subrahmanian 2010; Zimmermann et al. 2012).

In this paper, we consider RDF data enriched with trust information, and we study specific reasoning problems that arise within this setting. Indeed, RDF links between resources can be unilaterally set and be part of any data source as in the spirit of Tim Berners Lee's words: "Anyone can say anything about any topic and publish it anywhere". However, the lack of central control on the publishing of RDF data on the Web can have an impact on their accuracy, and it therefore becomes essential to associate trust information to data and being able to reason about them. To this end, the starting point of our analysis is the trust model proposed by (Hartig 2009), who firstly advocated the need of a uniform way to rate the trustworthiness of RDF data and of mechanisms to access and use these ratings. In fact, the analysis of (Hartig 2009) has been mainly carried out from the conceptual viewpoint, so that a number of the properties of the trust model, related to semantic and complexity issues, remain unexplored. In this paper we fill the gap.

In more detail, we first propose a model-theoretic semantics for the trust model by (Hartig 2009). The semantics is based on the concept of *trust aggregation functions*, which are functions designed to aggregate trust values coming from different RDF triples (Section 3.1). We then analyze the computational properties exhibited by the proposed framework (Section 3.2), and it turns out that:

- Even simple reasoning problems, such as checking whether there is a model for a given RDF graph, become intractable, formally NP-hard, in presence of trust information for general trust aggregation functions.

Motivated by the above bad news, we focus on specific classes of trust aggregation functions computable via binary *trust operators*, in the spirit of the general framework for annotated RDF data by (Zimmermann et al. 2012) and some proposals in relational (Karvounarakis and Green 2012) and RDF databases (Buneman and Kostylev 2010) where commutative semirings are considered. Within these classes, we analyze the concepts of *entailment*, *equivalence*, and *core* as suitable adaptations of corresponding concepts for RDF data (Gutierrez et al. 2011) and we study their intrinsic computational complexity (Section 4). It turns out that:

- The syntactic restriction introduced by trust operators is a key for the tractability of the reasoning problems in presence of trust information.
- While *entailment*, *equivalence*, and *core* are grounded on the semantics of the representation language, tight syntactic characterizations can be exhibited for them.
- Islands of tractability for these concepts can be singled out based on the structural properties of the interactions among the data. In particular, tractability can be established for acyclic or nearly-acyclic RDF graphs.

Finally, we point out that the trust framework and all the algorithms proposed in the paper have been implemented

*V. Fionda's work was supported by the European Commission, the European Social Fund and the Calabria region. G. Greco's work was also supported by a Kurt Gödel Research Fellowship, awarded by the Kurt Gödel Society.

and are made available in a prototype system.[1] We conducted experiments on real data in order to assess the applicability of the approach and the effectiveness of the algorithmic solutions. Implementation issues and experimental results are also discussed in the paper (Section 5).

2 Preliminaries

RDF Graphs. Let $\mathcal{U}$ and $\mathcal{B}$ be two disjoint infinite sets of *URI references* and *blank node identifiers*, respectively. An *RDF triple* has the form (s, p, o), where $s \in \mathcal{U} \cup \mathcal{B}$ is the *subject*, $p \in \mathcal{U}$ is the *property* (also called *predicate*), and $o \in \mathcal{U} \cup \mathcal{B}$ is the *object*.[2] The intuitive meaning is that the property p holds between the subject s and the object o (Hayes and Patel-Schneider 2014). A set $\mathcal{G}$ of RDF triples is usually called an RDF *graph*. Indeed, an RDF triple can be viewed as a directed edge, from the subject to the object nodes, labeled with the corresponding property (i.e., URI reference). In the paper, we deal with *simple* RDF graphs only, i.e., we do not specify any ad-hoc semantics for the properties occurring in the triples of $\mathcal{G}$.

Semantics of RDF graphs. An *interpretation* $\mathcal{I}$ over $\mathcal{U}$ is a tuple $\langle Res, Prop, PVal, Int\rangle$ where: (i) Res is a non-empty set of resources; (ii) $Prop$ is a non-empty set of property names, not necessarily disjoint from Res; (iii) $PVal : Prop \rightarrow 2^{Res \times Res}$ is a function that assigns a subset of $Res \times Res$ to each property name in $Prop$; and (iv) $Int : \mathcal{U} \rightarrow Res \cup Prop$ is an interpretation mapping that assigns a resource or property name to each element of $\mathcal{U}$.

$\mathcal{I}$ is a *model* of an RDF graph $\mathcal{G}$ if there is a blank node assignment $A : \mathcal{B} \rightarrow Res$, in the following referred to as the *witness* of $\mathcal{I}$, such that: for each triple $(s, p, o) \in \mathcal{G}$, $Int(p) \in Prop$ and $(Int_A(s), Int_A(o)) \in PVal(Int(p))$, where $Int_A(x) = Int(x)$ if $x \in \mathcal{U}$, and $Int_A(x) = A(x)$ if $x \in \mathcal{B}$. Note that blank nodes are interpreted existentially via the underlying witness A.

3 Trust Framework

In this section, we illustrate the trust framework to enrich standard RDF graphs with trust values, inspired by the approach proposed by (Hartig 2009). We also define its model-theoretic semantics and study its complexity. In the following, if h is a function, we denote by $\mathtt{dom}(h)$ its domain and, by slightly abusing notation, for each element $z \notin \mathtt{dom}(h)$, we write $h(z)=\phi$, where ϕ is a distinguished symbol acting as a neutral element. If $\langle z_1, .., z_m\rangle$ is a tuple of values, then $\langle h(z_1), .., h(z_m)\rangle$ is shortened to $h(\langle z_1, ..., z_m\rangle)$.

3.1 Syntax and Semantics of t-Graphs

The *trustworthiness* of an RDF triple $t = (s, p, o)$ is a value indicating to what extent t is believed or disbelieved to be true. A trust-enriched RDF graph (short: *t-graph*) is a pair $\langle \mathcal{G}, w\rangle$ where $\mathcal{G}$ is an RDF graph, and w is a real-valued *trust function* such that:

[1] See http://trdfreasoner.wordpress.com

[2] *Literals* are not considered, as they do not play a role in the formal analysis at the given abstraction level (Gutierrez et al. 2011).

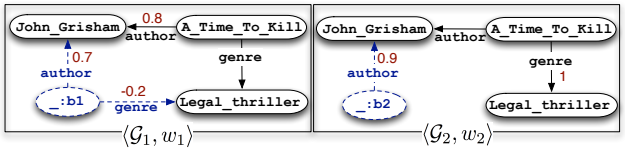

Figure 1: Examples of t-graphs.

- $\mathtt{dom}(w)$ is a set of RDF triples; here, the symbol ϕ associated with any triple outside the domain is meant to denote that the trust value of t is unknown;
- for each $t \in \mathtt{dom}(w)$, $-1 \leq w(t) \leq 1$ holds. Intuitively, the value 1 (resp., -1) represents the maximum belief (resp., disbelief) in the information encoded by the triple. A value of 0 represents absolute uncertainty on the reliability of the corresponding triple.

Note that $\mathtt{dom}(w) \supseteq \mathcal{G}$ is not enforced to hold. So, a tuple $t \in \mathcal{G}$ might have no associated trust value. In particular, $w(t) = \phi$ is different from $w(t) = 0$, as discussed by (Hartig 2009).

Example 1 Two t-graphs are illustrated in Figure 1. For instance, in $\langle \mathcal{G}_1, w_1\rangle$ we have that $w_1((\mathtt{A\_Time\_To\_Kill}, \mathtt{author}, \mathtt{John\_Grisham}))=0.8$ and $\mathtt{dom}(w_1) = \mathcal{G}_1 \setminus \{(\mathtt{A\_Time\_To\_Kill}, \mathtt{genre}, \mathtt{Legal\_thriller})\}$. ◁

In many cases, we need to aggregate the trust values that come from different RDF triples, even possibly from the entire RDF graph. To this end, we consider *trust aggregation functions*, i.e., functions taking as input a multiset (i.e., a set where repetitions are allowed) of elements in $[-1, 1] \cup \{\phi\}$ and producing an aggregate value in $[-1, 1] \cup \{\phi\}$. The specific choice of the trust aggregation function f depends on the requirements of the application at hand. However, as ϕ is a neutral element with respect to f, we consistently assume that $f(S) = f(S \cup \{\phi\})$ holds, for each S.[3]

As a concrete choice, we might aggregate values by taking the minimum one, if a *cautious* function $f_{\min}$ is used where the trust value of a set of triples is equal to the trust value of the least trusted triple. The *brave* perspective is instead given by the function $f_{\max}$ taking the maximum value.

Example 2 Consider again $\langle \mathcal{G}_1, w_1\rangle$ in Figure 1. According to the *cautious* perspective, we have $f_{\min}(\{w_1(t) \mid t \in \mathcal{G}_1\})= \min\{w_1(t) \mid t \in \mathcal{G}_1\}$=-0.2. According to the *brave* perspective, $f_{\max}(\{w_1(t) \mid t \in \mathcal{G}_1\})= \max\{w_1(t) \mid t \in \mathcal{G}_1\}$=0.8. ◁

The trust framework illustrated so far is essentially taken from (Hartig 2009), where however a formal model-theoretic semantics is not proposed. Our first contribution is to extend the concept of model to t-graphs in a way that it is parametric w.r.t. the trust aggregation function f.

Definition 1 (f-models) Let $\mathcal{I} = \langle Res, Prop, PVal, Int\rangle$ be an interpretation, and let $\bar{\sigma} = \sigma_{y|y \in Prop}$ be a family of real-valued functions with $\mathtt{dom}(\sigma_y) \subseteq PVal(y)$, for each $y \in Prop$, i.e., σ_y assigns trust values to pairs $(s, o) \in PVal(y)$. The pair $\langle \mathcal{I}, \bar{\sigma}\rangle$ is an *f-model* of a t-graph $\langle \mathcal{G}, w\rangle$ if there is a blank node assignment $A : \mathcal{B} \rightarrow Res$ such that:

[3] Operations over sets are hereinafter transparently applied to multisets, by taking multiplicity of elements into account.

(1) $\mathcal{I}$ is a model of $\mathcal{G}$, with A being the associated witness;
(2) for each $(s,p,o) \in \mathcal{G}$, $\sigma_{Int(p)}(Int_A(s), Int_A(o)) = f(\{\sigma_{Int(p)}(Int_A(s), Int_A(o))\} \cup S)$, where

$$S = \bigcup_{\substack{(s',p',o') \in \mathcal{G} \text{ such that} \\ Int_A((s,p,o)) = Int_A((s',p',o'))}} \{w((s',p',o'))\}.$$

□

Note that, according to our approach, standard interpretations are equipped with real-valued functions that provide an interpretation to trust values. Indeed, condition (1) is the standard definition of model for RDF graphs, while condition (2) is specific for trust values. In order to get an intuition, observe that, for the maximization function, condition (2) prescribes that the value assigned via $\bar{\sigma}$ to each "interpreted" triple in any model has to be an upper bound for the trust values of the RDF triples that are mapped into it.

Example 3 Consider again $\langle \mathcal{G}_1, w_1 \rangle$ in Figure 1. Let $\langle \mathcal{I}, \bar{\sigma} \rangle$ be the pair such that: $\sigma_p(s,o) = w_1((s,p,o))$, $\forall (s,p,o) \in \mathtt{dom}(w_1)$; $\sigma_p(s,o)=0$, $\forall (s,p,o) \in \mathcal{G}_1 \setminus \mathtt{dom}(w_1)$; and where $\mathcal{I} = \langle Res, Prop, PVal, Int \rangle$ is an interpretation whose resources and property names directly correspond to URI references and properties, respectively:

- Res={John_Grisham, author, A_Time_To_Kill, Legal_thriller, genre};
- $Prop$={author, genre},
- $PVal$(author)={(A_Time_To_Kill, John_Grisham)}, $PVal$(genre)= {(A_Time_To_Kill, Legal_thriller)};
- Int is the identity function;

Note that $\langle \mathcal{I}, \bar{\sigma} \rangle$ is an $f_{\max}$-model of the t-graph $\langle \mathcal{G}_1, w_1 \rangle$ in Figure 1, whose witness is the blank node assignment A with A(_:b1)=A_Time_To_Kill. Instead, observe that $\langle \mathcal{I}, \bar{\sigma} \rangle$ is not an $f_{\min}$-model. Indeed, we have that $\sigma_{\mathtt{genre}}$(A_Time_To_Kill, Legal_thriller)=0 and since A is the only possible blank node assignment, we obtain that $\min\{\sigma_{\mathtt{genre}}$(A_Time_To_Kill, Legal_thriller), w_1(A_Time_To_Kill, genre, Legal_thriller), w_1(_:b1, genre, Legal_thriller)$\}$=min$\{0, \phi$,-0.2$\}$= -0.2 $\neq$ 0 ◁

3.2 Complexity Analysis

The framework illustrated so far is general enough to fit a number of application domains. However, there is a price to be paid for this generality, which comes in the form of the intrinsic complexity of the basic reasoning problem arising with the concept of f-model. In fact, it is well-known and easy to see that, given an interpretation $\mathcal{I}$ and an RDF graph $\mathcal{G}$, deciding whether $\mathcal{I}$ is a model of $\mathcal{G}$ is NP-complete in general, but it is feasible in polynomial time if $\mathcal{G}$ is acyclic. Opposed to this goods news, it turns out that deciding whether $\langle \mathcal{I}, \bar{\sigma} \rangle$ is an f-model of $\langle \mathcal{G}, w \rangle$ (according to Definition 1) is intractable even over acyclic graphs.

Formally, let CHECK$_{f,\mathcal{C}}$ be the problem receiving as input a pair $\langle \mathcal{I}, \bar{\sigma} \rangle$ and a t-graph $\langle \mathcal{G}, w \rangle$ with $\mathcal{G}$ belonging to the class $\mathcal{C}$ of RDF graphs, and asking to decide whether $\langle \mathcal{I}, \bar{\sigma} \rangle$ is an f-model of $\langle \mathcal{G}, w \rangle$. Then, the following holds.

Theorem 1 *Assume that f is a polynomial-time computable function. Then,* CHECK$_{f,\mathcal{C}}$ *is* NP*-complete. Hardness holds even if $\mathcal{C}$ is the class of all acyclic RDF t-graphs.*

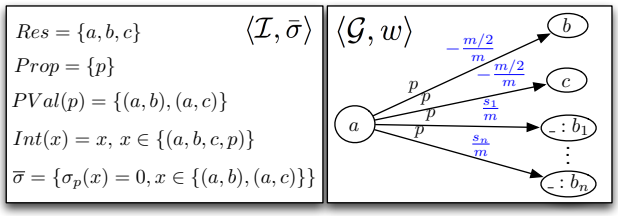

Figure 2: Example construction in the proof of Theorem 1.

Proof Sketch. NP-hardness can be shown via a reduction from the PARTITION problem of deciding whether there is a way to partition a multiset $S = \{s_1, s_2, \ldots, s_n\}$ of integers into two multisets S_1 and S_2 such that the sum of the numbers in S_1 equals the sum of the numbers in S_2.

Based on S, we build the f-model checking instance $(\langle \mathcal{I}, \bar{\sigma} \rangle, \langle \mathcal{G}, w \rangle)$ reported in Figure 2 with f being the average function, where blank nodes _:$b_i \in \mathcal{G}$ are in one-to-one correspondence with the numbers $s_i \in S$, and where the trust values (normalized by the factor $m = \sum_{s_i \in S} s_i$) are given by the weights associated with the edges. Eventually, one can check that $\langle \mathcal{I}, \bar{\sigma} \rangle$ is an f-model of $\langle \mathcal{G}, w \rangle$ if, and only if, the answer to the PARTITION instance is positive. Indeed, $\langle \mathcal{I}, \bar{\sigma} \rangle$ is a model if, and only if, blank nodes (hence numbers) can be partitioned in two groups, mapped to nodes b and c, respectively, whose trust values sum up to $1/2$. □

4 Trust Aggregation Operators

In order to circumvent the above intractability, we have to focus on specific kinds of trust aggregation functions. Our choice is to consider functions that can be built on top of binary operators, enjoying algebraic properties which are very often considered in the literature.

Definition 2 (trust operator) A *trust (aggregation) operator* $\oplus$ is a binary operator defined over $[-1,1] \cup \{\phi\}$ and satisfying the following axioms:

1. $\oplus$ is a binary closed operator, i.e., $\oplus : [-1,1] \cup \{\phi\} \times [-1,1] \cup \{\phi\} \mapsto [-1,1] \cup \{\phi\}$;
2. $\oplus$ is associative, i.e., $\forall v, v', v'' \in [-1,1] \cup \{\phi\}$, the equation $(v \oplus v') \oplus v'' = v \oplus (v' \oplus v'')$ holds;
3. ϕ is the identity element, i.e., $\forall v \in [-1,1] \cup \{\phi\}$, the equation $v \oplus \phi = \phi \oplus v = v$ holds.
4. $\preceq$, defined as $v \preceq v'$ if, and only if, $v \oplus v' = v$, is a partial order over $([-1,1] \cup \{\phi\}, \oplus)$ that is compatible with $\oplus$, i.e., if $v \preceq v'$ then $v \oplus v'' \preceq v' \oplus v''$, $\forall v, v', v'' \in [-1,1] \cup \{\phi\}$;

Therefore, $([-1,1] \cup \{\phi\}, \oplus, \preceq)$ is an Ordered Monoid with the partial order $\preceq$. Moreover, we require that the following two additional axioms are satisfied:

5. $\oplus$ is idempotent, i.e., $\forall v \in [-1,1] \cup \{\phi\}$, $v \oplus v = v$ (in fact, this means that $\preceq$ is reflexive);
6. $\oplus$ is commutative, i.e., $\forall v, v' \in [-1,1] \cup \{\phi\}$, the equation $v \oplus v' = v' \oplus v$ holds. □

In the following, the symbol "$\oplus$" will always refer to a trust operator. A derived property of "$\oplus$" is stated below.

Proposition 1 (decomposability) *Let v, v', v'' be in $[-1,1] \cup \{\phi\}$. Then, $v \preceq v' \oplus v'' \Leftrightarrow v \preceq v'$ and $v \preceq v''$.*

Proof. ($\Rightarrow$) Observe that $v \preceq v' \oplus v''$ implies $v = v \oplus v' \oplus v''$. Since $\preceq$ is compatible with $\oplus$, we have $v \oplus v' \preceq v \oplus v' \oplus v'' \oplus v'$, which implies $v \oplus v' = v \oplus v' \oplus v \oplus v' \oplus v'' \oplus v'$. By commutativity and idempotence, we obtain $v \oplus v' = v \oplus v' \oplus v''$, that is, $v \oplus v' = v$ and thus $v \preceq v'$. A similar reasoning applies to show that $v \preceq v''$.

($\Leftarrow$) Since $\preceq$ is compatible with $\oplus$, $v \preceq v'$ (resp., $v \preceq v''$) implies $v \oplus v'' \preceq v' \oplus v''$ (resp., $v \oplus v \preceq v'' \oplus v$). By commutativity and idempotence, we obtain $v = v \oplus v \preceq v \oplus v''$. Then, by transitivity, we derive $v \preceq v' \oplus v''$. □

A trust operator $\oplus$ naturally induces the trust aggregation function $f_\oplus$ such that, for each multiset S of elements in $[-1,1] \cup \{\phi\}$, $f_\oplus(S) = \oplus_{v \in S}\, v$. Note that this is well-defined, as $\oplus$ is commutative and associative (so the order is immaterial). Moreover, note that $\min$ and $\max$ are trust operators. Hence, the notation $f_{\min}$ and $f_{\max}$ used in the previous section is consistent. Now, we claim that, by dealing with trust operators, no unexpected complexity blow up occurs. The result is a simple adaptation of a stronger technical result discussed in Theorem 6. Here, we just stress that the average aggregation function is not expressible via binary operators, so that the result below does not apply to it—indeed, we already know from the proof of Theorem 1 that this operator quickly leads to intractability.

Theorem 2 *If $\mathcal{A}$ is a class of acyclic RDF graphs, then* $\textsc{Check}_{f_\oplus, \mathcal{A}}$ *is feasible in polynomial time.*

Now that trust operators have been shown to be computationally well-designed, we proceed to analyze their properties. In particular, we shall next focus on concepts playing a relevant role in the analysis of RDF redundancies. Before doing so, we point out that the specialization of Definition 1 to a function $f_\oplus$, for any given trust operator $\oplus$, nicely fits the concept of model designed by (Zimmermann et al. 2012) for the framework of annotated RDF graphs–in its turn inspired by (Kifer and Subrahmanian 1992). In fact, it can be checked that our results smoothly apply to that general framework for annotated RDF graphs, too.

More abstractly, our framework is also related to a number of works where commutative *semirings* are used to record the provenance of query results when dealing with annotated data (Karvounarakis and Green 2012; Buneman and Kostylev 2010). With this respect, note that semirings have two different kinds of operations: *(i)* the addition operation (corresponding to our aggregation function) that is used to combine annotations (in our case, trust values) referring to the same tuple; *(ii)* the product operation that is used to join annotations of different tuples in the context of *query answering* (more generally, reasoning about RDF schema). As we do not care about querying mechanisms (or RDF schema reasoning) in the paper, it is enough to focus on the addition operation only. In fact, entailment check and core computation on RDF data with annotated triples has been not considered in such related works, and our results smoothly apply to the semiring-based models, too.

4.1 Entailment for t-Graphs

We start the analysis by presenting a generalization of the concept of *entailment* for t-graphs.

Definition 3 ($\oplus$-entailment) Let $\langle \mathcal{G}_1, w_1 \rangle$ and $\langle \mathcal{G}_2, w_2 \rangle$ be two t-graphs. Then, $\langle \mathcal{G}_1, w_1 \rangle$ *$\oplus$-entails* $\langle \mathcal{G}_2, w_2 \rangle$ (denoted by $\langle \mathcal{G}_1, w_1 \rangle \models \langle \mathcal{G}_2, w_2 \rangle$, if $\oplus$ is understood) if every $f_\oplus$-model of $\langle \mathcal{G}_1, w_1 \rangle$ is also a $f_\oplus$-model of $\langle \mathcal{G}_2, w_2 \rangle$. □

Entailment has been studied for RDF graphs (without trust functions) and syntactic characterizations are known for it (Gutierrez et al. 2011). So, it is natural to look for similar characterizations in our context. To this end, we say that a function $\mu : \mathcal{U} \cup \mathcal{B} \to \mathcal{U} \cup \mathcal{B}$ is a URI preserving *mapping* if $\mu(u) = u$, for each $u \in \mathcal{U}$ as defined in (Gutierrez et al. 2011). For an RDF graph $\mathcal{G}$, let $\mu(\mathcal{G})$ be the set of triples $\{\mu(t) \mid t \in \mathcal{G}\}$ resulting from the application of μ. By exploiting the results of (Gutierrez et al. 2011) in our context, the following can be obtained.

Theorem 3 *Let w_ϕ be the function such that* $\texttt{dom}(w_\phi) = \emptyset$. *Then, $\langle \mathcal{G}_1, w_\phi \rangle \models \langle \mathcal{G}_2, w_\phi \rangle$ if, and only if, there is a URI preserving mapping μ with $\mu(\mathcal{G}_2) \subseteq \mathcal{G}_1$.*

Proof Sketch. Let $\langle \mathcal{I}, \bar{\sigma} \rangle$ be an interpretation and $\mathcal{G}$ an RDF graph. If $\langle \mathcal{I}, \bar{\sigma} \rangle$ is an $f_\oplus$-model of $\langle \mathcal{G}, w_\phi \rangle$, then for each triple $(s,p,o) \in \mathcal{G}$, $\sigma_{Int(p)}(Int_A(s), Int_A(o)) \preceq \phi$ must hold, being ϕ the identity element. Thus, Definition 1 reduces to the standard concept of model for RDF graphs, i.e., $\oplus$ plays no role. The result then follows by (Gutierrez et al. 2011). □

To extend the result to arbitrary trust functions, we need to introduce some restrictions on the allowed mappings, which are meant to recast Definition 1 in purely syntactic terms.

Definition 4 ($\oplus$-preserving mappings) Let $\langle \mathcal{G}_1, w_1 \rangle$ and $\langle \mathcal{G}_2, w_2 \rangle$ be two t-graphs. A mapping $\mu : \mathcal{U} \cup \mathcal{B} \to \mathcal{U} \cup \mathcal{B}$ is *$\oplus$-preserving for $\langle \mathcal{G}_1, w_1 \rangle$ w.r.t. $\langle \mathcal{G}_2, w_2 \rangle$* if μ is URI preserving, $\mu(\mathcal{G}_2) \subseteq \mathcal{G}_1$ and, for each $t \in \mathcal{G}_2$, $w_1(\mu(t)) \preceq w_2(t)$. □

Example 4 Consider again the two t-graphs in Figure 1. If μ is a URI preserving mapping such that $\mu(\texttt{\_:b2}) = \texttt{\_:b1}$ or $\mu(\texttt{\_:b2}) = \texttt{A\_Time\_To\_Kill}$, then μ is min-preserving for $\langle \mathcal{G}_1, w_1 \rangle$ w.r.t. $\langle \mathcal{G}_2, w_2 \rangle$. ◁

We next show that the entailment for t-graphs is intimately related with the existence of $\oplus$-preserving mappings.

Theorem 4 *$\langle \mathcal{G}_1, w_1 \rangle$ $\oplus$-entails $\langle \mathcal{G}_2, w_2 \rangle$ if, and only if, there is a $\oplus$-preserving mapping for $\langle \mathcal{G}_1, w_1 \rangle$ w.r.t. $\langle \mathcal{G}_2, w_2 \rangle$.*

Proof Sketch. Assume that $\langle \mathcal{G}_1, w_1 \rangle \models \langle \mathcal{G}_2, w_2 \rangle$, and consider the Herbrand interpretation (Hayes and Patel-Schneider 2014) $\langle \mathcal{H} = \langle Res, Prop, PVal, Int \rangle, \bar{\sigma} \rangle$ of $\mathcal{G}_1$, such that for each $(s,p,o) \in \mathcal{G}_1$, $\sigma_p(s,o) = w_1((s,p,o))$. By construction, $\langle H, \bar{\sigma} \rangle$ is an $f_\oplus$-model of $\langle \mathcal{G}_1, w_1 \rangle$ with the identity function $A_\mathcal{H}$ as witness. Since $\langle \mathcal{G}_1, w_1 \rangle \models \langle \mathcal{G}_2, w_2 \rangle$, $\langle \mathcal{H}, \bar{\sigma} \rangle$ is an $f_\oplus$-model of $\langle \mathcal{G}_2, w_2 \rangle$, for some witness A'. Let μ be the mapping such that $\mu(x) = Int_{A'}(x)$. All the triples of $\mathcal{G}_2$ are interpreted through μ as triples of $\mathcal{G}_1$ and, thus, $\mu(\mathcal{G}_2) \subseteq \mathcal{G}_1$. Moreover, one can check that μ is $\oplus$-preserving, due to the decomposability property (cf. Proposition 1).

Let now μ be a $\oplus$-preserving mapping and let $\langle \mathcal{I}, \bar{\sigma} \rangle$ be an $f_\oplus$-model of $\langle \mathcal{G}_1, w_1 \rangle$, whose witness is A. The blank

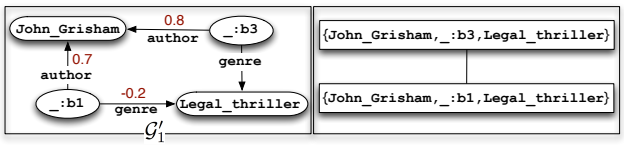

Figure 3: Structures in Example 6.

node assignment $B(x)$=$A(\mu(x))$ is a witness for $\langle \mathcal{I}, \bar{\sigma} \rangle$ being an $f_{\oplus}$-model of $\langle \mathcal{G}_2, w_2 \rangle$. □

Example 5 Recall from Example 4 that a min-preserving mapping exists for the t-graphs in Figure 1. Moreover, we claim that $\langle \mathcal{G}_1, w_1 \rangle \models \langle \mathcal{G}_2, w_2 \rangle$ holds. Indeed, whatever $f_{\min}$-model $\langle I, \bar{\sigma} \rangle$ of $\langle \mathcal{G}_1, w_1 \rangle$ must be such that $\min\{\sigma_{Int(\texttt{author})}(\texttt{A\_Time\_To\_Kill}, \texttt{John\_Grisham}), w_1((\texttt{A\_Time\_To\_Kill}, \texttt{author}, \texttt{John\_Grisham}))\} \leq 0.8$. So, $\langle I, \bar{\sigma} \rangle$ is also an $f_{\min}$-model of $\langle \mathcal{G}_2, w_2 \rangle$ with witness the blank node assignment $A(\texttt{\_:b2})$=$\texttt{A\_Time\_To\_Kill}$. ◁

Note that Theorem 4 generalizes the syntactic characterization of entailment defined for RDF graphs (Gutierrez et al. 2011) in presence of trust information. Besides its own conceptual relevance, the result plays an important role for a deeper analysis of the complexity of the setting.

Let $\oplus$-ENTAILMENT be the problem receiving as input a pair $\langle \mathcal{G}_1, w_1 \rangle$ and $\langle \mathcal{G}_2, w_2 \rangle$ and asking whether $\langle \mathcal{G}_1, w_1 \rangle$ $\oplus$-entails $\langle \mathcal{G}_2, w_2 \rangle$. Since the size of $\oplus$-preserving mappings is polynomially bounded, the following is easily seen to hold,[4] with the hardness trivially deriving from the corresponding one over RDF graphs (Gutierrez et al. 2011).

Theorem 5 $\oplus$-ENTAILMENT *is* NP*-complete.*

Note that Theorem 5 tells us that trust operators provide a method to deal with trust values without representing a source of additional complexity w.r.t. the basic RDF setting.

However, the fact that even the basic setting of RDF graphs without trust values is intractable is clearly not satisfying in general, and motivates the analysis of special classes of RDF and t-graphs. In particular, as often done in the literature, we next focus on nearly-acyclic graphs as they can be formalized via the concept of *tree decomposition*.

Definition 5 A tree decomposition of an RDF graph $\mathcal{G}$ is a pair $\langle T, \chi \rangle$, where $T = (V, E)$ is a tree and χ is a labeling function associating each vertex $v \in V$ with a set of nodes of $\mathcal{G}$ such that for each triple $(s, p, o) \in \mathcal{G} \setminus \{(s, p, o) \in \mathcal{G} \mid \{s, p, o\} \subseteq \mathcal{U}\}$, i.e., for each triple involving blank nodes, the following properties hold:

- there is a vertex $v \in V$ such that $\chi(v) \supseteq \{s, o\}$; and
- the sets of vertices $\{v \in V \mid s \in \chi(v)\}$ and $\{v \in V \mid o \in \chi(v)\}$ induce two connected subtrees over T.

The *width* of $\langle T, \chi \rangle$ is the value $\min_{v \in V} |\chi(v)| - 1$. The *treewidth* of an RDF graph $\mathcal{G}$, denoted by $tw(\mathcal{G})$, is the minimum width over all its possible tree decompositions. □

[4] In the results, we assume the usual computation model with unit costs for all operations involving numbers (in particular, for their manipulation via the $\oplus$ operator).

Note that edge orientation is immaterial in the above notion. Moreover, triples that do not contain blank nodes do not play any role no matter of their interconnections. As an example, the two graphs in Figure 1 have both treewidth 1, i.e., they are acyclic, while the undirected version of $\mathcal{G}_1$ (including the triples without blank nodes) contains a cycle.

Example 6 Consider the graph $\mathcal{G}'_1$ shown on the left of Figure 3, which is obtained from $\mathcal{G}_1$ by just replacing the resource $\texttt{A\_Time\_To\_Kill}$ with a blank node. The (undirected version of the) graph contains a cycle over edges including blank nodes. Indeed, its treewidth is 2, as it is witnessed by the tree decomposition shown on the right. ◁

Let $\oplus$-ENTAILMENT$_{\mathcal{C}}$ be the restriction of the problem $\oplus$-ENTAILMENT where $\mathcal{G}_2$ (but not necessarily $\mathcal{G}_1$) belongs to a given class $\mathcal{C}$. We next show that this problem is tractable over classes of bounded treewidth, hence generalizing the tractability of RDF graphs (without trust values) having bounded treewidth (Pichler et al. 2008). Hereinafter, $\mathcal{T}_k$ is the class of RDF graphs whose treewidth is k at most.

Theorem 6 *Let $k > 0$ be a fixed natural number. Then, $\oplus$-ENTAILMENT$_{\mathcal{T}_k}$ is computable in polynomial time.*

Proof Sketch. Given two t-graphs $\langle \mathcal{G}_1, w_1 \rangle$ and $\langle \mathcal{G}_2, w_2 \rangle$ such that $tw(\mathcal{G}_2)$=k, $\langle \mathcal{G}_1, w_1 \rangle$|=$\langle \mathcal{G}_2, w_2 \rangle$ can be checked in polynomial time with an adaptation of the algorithm presented in (Pichler et al. 2008). Let $\langle T, \chi \rangle$ be one of the tree decompositions of $\langle \mathcal{G}_2, w_2 \rangle$ with width k. T is visited according to a post-order depth-first traversal, i.e., from the leaves to the root, by associating to each node v the set of $\oplus$-preserving mappings μ such that $\mu(\mathcal{G}_v) \subseteq \mathcal{G}_1$, where $\mathcal{G}_v \subseteq \mathcal{G}_2$ contains all the triples of $\mathcal{G}_2$ whose subject or object belong to $\chi(v)$. The sets are propagated from the children to the parent via left semi joins. At the end, if the root has a non empty set of mappings, then $\langle \mathcal{G}_1, w_1 \rangle$|=$\langle \mathcal{G}_2, w_2 \rangle$. □

As a specialization of the above result, we get that $\oplus$-ENTAILMENT$_{\mathcal{C}}$ is tractable on any class $\mathcal{C}$ of graphs that do not involve at all blank nodes—so that the notion of treewidth trivializes because $\mathcal{G} \setminus \{(s, p, o) \in \mathcal{G} \mid \{s, p, o\} \subseteq \mathcal{U}\} = \emptyset$, for each $\mathcal{G} \in \mathcal{C}$. Note that, unlike the standard setting of RDF graphs, even this basic result is not immediate in our setting because of the presence of trust functions.

4.2 Cores of t-Graphs

Another concept we would like to explore is the *core*. Indeed, when reasoning about RDF data redundancy, it plays a fundamental role by providing a "normal form" for graphs. Before introducing the concept of core of a t-graph, let us introduce a related notion.

Definition 6 A t-graph $\langle \mathcal{G}, w \rangle$ is $\oplus$*-lean* if there is no $\oplus$-preserving mapping for $\langle \mathcal{G}', w \rangle$ w.r.t. $\langle \mathcal{G}, w \rangle$, $\forall \mathcal{G}' \subset \mathcal{G}$. □

We are now ready to define the concept of $\oplus$-core.

Definition 7 ($\oplus$-core) Let $\langle \mathcal{G}, w \rangle$ be a t-graph. A $\oplus$*-core* of $\langle \mathcal{G}, w \rangle$ is a t-graph $\langle \mathcal{G}', w \rangle$ with $\mathcal{G}' \subseteq \mathcal{G}$ and such that: *(i)* there is a $\oplus$-preserving mapping for $\langle \mathcal{G}', w \rangle$ w.r.t. $\langle \mathcal{G}, w \rangle$; and *(ii)* $\langle \mathcal{G}', w \rangle$ is $\oplus$-lean. □

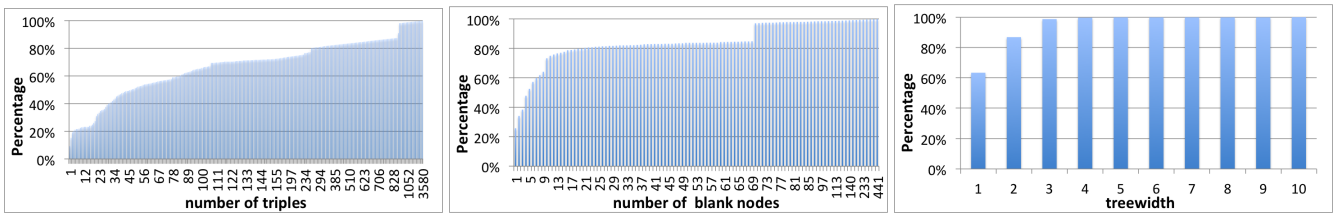

Figure 4: Distribution of RDF graphs in the dataset.

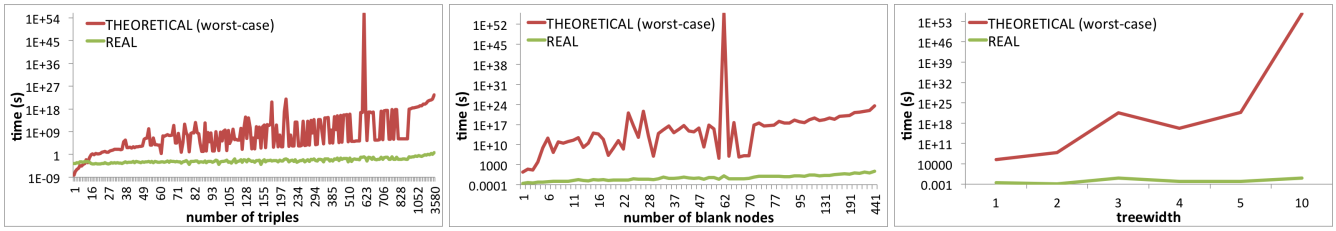

Figure 5: Running time of $\oplus$-entailment checking.

Note that the above definition reduces to the standard one for RDF graphs (Hayes and Patel-Schneider 2014), by getting rid of the function w. In fact, it is well-known that every RDF graph $\mathcal{G}$ has a core, that all its cores are *equivalent* (in fact, *isomorphic*), and that any core of $\mathcal{G}$ is equivalent to $\mathcal{G}$ itself. It is immediate to see that the latter property is preserved in presence of trust values.

Proposition 2 *If $\langle \mathcal{G}', w \rangle$ is a $\oplus$-*core *of $\langle \mathcal{G}, w \rangle$, then $\langle \mathcal{G}', w \rangle$ and $\langle \mathcal{G}, w \rangle$ are $\oplus$-equivalent, i.e., $\oplus$-entail each other.*

Proof. By Theorem 4, $\langle \mathcal{G}', w \rangle$ and $\langle \mathcal{G}, w \rangle$ are $\oplus$-equivalent if, and only if, there is a $\oplus$-preserving mapping μ for $\langle \mathcal{G}, w \rangle$ w.r.t. $\langle \mathcal{G}', w \rangle$ and a $\oplus$-preserving mapping μ' for $\langle \mathcal{G}', w \rangle$ w.r.t. $\langle \mathcal{G}, w \rangle$. The existence of μ' is guaranteed by point *(i)* in Definition 7, whereas μ can be just the identity function. □

In order to show that the other properties are preserved as well, we need some elaborations.

Theorem 7 *Every t-graph $\langle \mathcal{G}, w \rangle$ has a $\oplus$-core. Moreover, any two cores of $\langle \mathcal{G}, w \rangle$ are $\oplus$-equivalent.*

Proof Sketch. Let $\langle \mathcal{G}', w \rangle$ and $\langle \mathcal{G}'', w \rangle$ be two $\oplus$-cores of $\langle \mathcal{G}, w \rangle$ and μ' (resp., μ'') be the $\oplus$-preserving mapping for $\langle \mathcal{G}', w \rangle$ (resp., $\langle \mathcal{G}'', w \rangle$) w.r.t. $\langle \mathcal{G}, w \rangle$. Since $\oplus$ is decomposable and $\mathcal{G}' \subseteq \mathcal{G}$, μ'' is also $\oplus$-preserving for $\langle \mathcal{G}'', w \rangle$ w.r.t. $\langle \mathcal{G}', w \rangle$. Because of Theorem 4, we have $\langle \mathcal{G}', w \rangle \models \langle \mathcal{G}'', w \rangle$. Symmetrically, we could derive that $\langle \mathcal{G}'', w \rangle \models \langle \mathcal{G}', w \rangle$, hence the cores are $\oplus$-equivalent. As for the existence, let $\mu_1, ..., \mu_n$ be a maximal sequence such that μ_i, for each $i \in \{1, ..., n-1\}$, is $\oplus$-preserving for $\langle \mathcal{G}_{i+1}, w \rangle$ w.r.t. $\langle \mathcal{G}_i, w \rangle$, where $\mathcal{G}_{i+1} = \mu_i(\mathcal{G}_i) \subset \mathcal{G}_i$ and $\mathcal{G}_1 = \mathcal{G}$. Such a sequence is finite because it is a succession of contractions of the original graph. It can be shown that the composition $\circ$ of two $\oplus$-preserving mappings is $\oplus$-preserving and thus, the composition $\mu^* = \mu_1 \circ ... \circ \mu_n$ is $\oplus$-preserving for $\langle \mathcal{G}_n, w \rangle$ w.r.t. $\langle \mathcal{G}, w \rangle$, where $\mathcal{G}_n = \mu^*(\mathcal{G}) \subseteq \mathcal{G}$. By the maximality of the sequence, $\langle \mathcal{G}_n, w \rangle$ is lean, and hence it is a $\oplus$-core. □

Note that the proof of the existence of a core is actually constructive. This is the key to establish the following result.

Theorem 8 *Let k be a fixed natural number. A $\oplus$-core of any $\langle \mathcal{G}, w \rangle$ with $\mathcal{G} \in \mathcal{T}_k$ can be computed in polynomial time.*

Proof Sketch. The $\oplus$-core can be computed via an adaptation of the method in (Gottlob and Nash 2008) by iteratively trying to contract $\mathcal{G}$ by removing one blank node and all the triples involving it (thus obtaining a proper subgraph $\mathcal{G}'$ of $\mathcal{G}$) and by checking in polynomial time (by Theorem 6) if $\langle \mathcal{G}, w \rangle$ $\oplus$-entails $\langle \mathcal{G}', w \rangle$. After all blank nodes have been checked, the last valid contraction $\langle \mathcal{G}', w \rangle$ is the $\oplus$-core. □

5 Implementation and Evaluation

The trust framework has been implemented as an extension of the Jena API (http://jena.apache.org) and can be downloaded at http://trdfreasoner.wordpress.com. It provides classes to represent and manage trust values associated to RDF statements and implements some trust aggregation functions. Both algorithms for $\oplus$-entailment checking and $\oplus$-core computation have been implemented by relying on the LibTW library (http://www.treewidth.com) to compute tree decompositions.

Evaluation. An empirical study on the usage of blank nodes in RDF data on the Web has been recently carried out by (Hogan et al. 2014) on the BTC dataset (Harth 2012), by showing that the 44.9% of RDF graphs feature at least one blank node and the majority of them contain acyclic (62.3%) or nearly-acyclic (37.7% with treewidth equals to 2) structures on which Theorem 6 and Theorem 8 apply. In our experiments, we have considered again the BTC dataset, by focusing on the 10 top publishers of blank nodes as reported in (Hogan et al. 2014) and by randomly sampling ∼1K RDF graphs from them. Figure 4 reports some statistics about the

resulting dataset D w.r.t. the number of triples, blank nodes and treewidth as cumulative bar charts.

For each RDF graph in D, we generated 5 t-graphs with random trust values, computed the $\oplus$-core of each of them and checked the $\oplus$-entailment between each pair. In addition, we computed the $\oplus$-core of each graph in D by considering all trust values equal to 1. Running times have been compared with the (worst-case) theoretical estimates that can be derived by analyzing the algorithms.[5] Results for $\oplus$-entailment checking are reported in logarithmic scale in Figure 5 where they are compared with the theoretical estimates—experiments have been executed on an PC Intel Core i7 2,8 GHz, 16GB RAM. Trends are almost identical for $\oplus$-core computation. Running times are reported w.r.t. the number of triples (nt), number of blank nodes (nb) and treewidth (tw). For each input configuration, we executed 5 runs and computed the running time as the average of the 3 running times obtained by excluding the lowest and highest ones. For each parameter $p \in \{nt, nb, tw\}$, D has been partitioned such that all the graphs in the same partition have the same value of p (e.g., as for tw, $\mathcal{G}_i$ and $\mathcal{G}_j$ belong to the same partition if, and only if, $tw(\mathcal{G}_i)=tw(\mathcal{G}_j)$). For each value of p, we computed the real and theoretical running times as the average on all the graphs of the corresponding partition. As can be noted both algorithms perform much faster than the worst-case theoretical bounds.

6 Discussion and Conclusions

The use of RDF helps improving productivity and efficiency in terms of data dissemination and integration, by automatizing processes with minimal human intervention. However, the presence of incorrect and unreliable data can negatively affect decision processes and cause economic damages. By associating trust values to RDF data, some of these issues can be mitigated. However, having trust values alone is not enough. Indeed, since RDF is the backbone of the Semantic Web, making reasoning problems tractable when dealing with such a large volume of data is essential. This paper contributed in this direction by defining a formal framework (and a prototype system) for reasoning about trust values and by singling out islands of tractability (for classes of acyclic and nearly-acyclic graphs) for the most basic problems arising therein.

References

Buneman, P., and Kostylev, E. 2010. Annotation algebras for RDFS. In *The Second International Workshop on the role of Semantic Web in Provenance Management (SWPM-10)*.

Dividino, R. Q.; Sizov, S.; Staab, S.; and Schueler, B. 2009. Querying for provenance, trust, uncertainty and other meta knowledge in RDF. *J. Web Sem.* 7(3):204–219.

Gottlob, G., and Nash, A. 2008. Efficient core computation in data exchange. *J. ACM* 55(2).

Gutierrez, C.; Hurtado, C. A.; Mendelzon, A. O.; and Prez, J. 2011. Foundations of Semantic Web databases. *J. Comput. Syst. Sci.* 77(3):520–541.

Gutierrez, C.; Hurtado, C.; and Vaisman, A. 2007. Introducing time into RDF. *IEEE Transactions on Knowledge and Data Engineering* 19(2):207–218.

Harth, A. 2012. Billion Triples Challenge data set. Downloaded from http://km.aifb.kit.edu/projects/btc-2012/.

Hartig, O. 2009. Querying Trust in RDF Data with tSPARQL. In *6th Annual European Semantic Web Conference (ESWC2009)*, 5–20.

Hayes, P., and Patel-Schneider, P. 2014. RDF 1.1 Semantics. W3C recommendation.

Hogan, A.; Arenas, M.; Mallea, A.; and Polleres, A. 2014. Everything you always wanted to know about blank nodes. *Web Semantics: Science, Services and Agents on the World Wide Web* In press.

Karvounarakis, G., and Green, T. J. 2012. Semiring-annotated data: queries and provenance? *SIGMOD Record* 41(3):5–14.

Kifer, M., and Subrahmanian, V. S. 1992. Theory of Generalized Annotated Logic Programming and its Applications. *J. Log. Program.* 12(3&4):335–367.

Pichler, R.; Polleres, A.; Wei, F.; and Woltran, S. 2008. dRDF: Entailment for Domain-Restricted RDF. In *ESWC*, volume 5021 of *Lecture Notes in Computer Science*, 200–214. Springer.

Straccia, U. 2009. A Minimal Deductive System for General Fuzzy RDF. In Polleres, A., and Swift, T., eds., *RR*, volume 5837 of *Lecture Notes in Computer Science*, 166–181. Springer.

Tomaszuk, D.; Pak, K.; and Rybinski, H. 2013. Trust in RDF Graphs. In Morzy, T.; Hrder, T.; and Wrembel, R., eds., *ADBIS*, volume 186 of *Advances in Intelligent Systems and Computing*, 273–283. Springer.

Udrea, O.; Recupero, D. R.; and Subrahmanian, V. S. 2010. Annotated RDF. *ACM Trans. Comput. Log.* 11(2).

Zimmermann, A.; Lopes, N.; Polleres, A.; and Straccia, U. 2012. A general framework for representing, reasoning and querying with annotated Semantic Web data. *Web Semantics: Science, Services and Agents on the World Wide Web* 11:72–95.

[5] In fact, both algorithms are easily seen to be polynomial in the size of the input graphs and exponential in the treewidth.