Satisfaction and Implication of Integrity Constraints in Ontology-based Data Access

Charalampos Nikolaou[1,2] , **Bernardo Cuenca Grau**[1] , **Egor V. Kostylev**[1] , **Mark Kaminski**[1] and **Ian Horrocks**[1]

[1]Department of Computer Science, University of Oxford, UK
[2]Infor, UK

Abstract

We extend ontology-based data access with integrity constraints over both the source and target schemas. The relevant reasoning problems in this setting are *constraint satisfaction*—to check whether a database satisfies the target constraints given the mappings and the ontology—and source-to-target (resp., target-to-source) *constraint implication*, which is to check whether a target constraint (resp., a source constraint) is satisfied by each database satisfying the source constraints (resp., the target constraints). We establish decidability and complexity bounds for all these problems in the case where ontologies are expressed in *DL-Lite*$_\mathcal{R}$ and constraints range from functional dependencies to disjunctive tuple-generating dependencies.

1 Introduction

Ontology-based data access (OBDA) is a popular approach for integrating heterogeneous data sources [Poggi *et al.*, 2008; Calvanese *et al.*, 2017; Kharlamov *et al.*, 2017]. In OBDA, an ontology represents a global conceptual model reconciling the mismatches between the source schemas and providing users with the vocabulary for query formulation. The relationship between the ontology and the source schemas is declaratively specified using global-as-view (GAV) mappings [Lenzerini, 2002], which assign to ontology predicates a set of views over the data sources.

Example 1. *Consider the integration of a cadastre database and a database recording housing benefit applications. The cadastre uses table* T(pname, owner, sqm) *to record information about properties, such as name (*pname*), ownership (*owner*), and area in square meters (*sqm*). The housing benefit authority uses table* S(aname, apr) *to record the name of the applicant and the relevant property. The following are sample instances of these tables.*

T:
pname	owner	sqm
22 Bauhaus	B. Lyndon	82

S:
aname	apr
M. Sim	22 Bauhaus
M. Bloom	30 Rossetti

Mappings (1)–(3) *below are used to populate the ontology concept* Property *and roles* heldBy *and* claims.

$$\text{T}(x,y,z) \rightarrow \text{Property}(x) \quad (1) \qquad \text{S}(x,y) \rightarrow \text{claims}(x,y) \quad (3)$$
$$\text{T}(x,y,z) \rightarrow \text{heldBy}(x,y) \quad (2)$$

Axiom (4) *below identifies anyone making an application as a* Claimant*, whereas* (5) *identifies anyone holding a property as a* Landlord*. Axioms* (6)–(7) *also state that everything claimed for by a claimant or held by a landlord is a property.*

$$\exists \text{claims} \sqsubseteq \text{Claimant} \quad (4) \qquad \exists \text{claims}^- \sqsubseteq \text{Property} \quad (6)$$
$$\exists \text{heldBy}^- \sqsubseteq \text{Landlord} \quad (5) \qquad \exists \text{heldBy} \sqsubseteq \text{Property} \quad (7)$$

When integrating data under a common schema, one may want to ensure that certain constraints are globally satisfied. In Example 1 we may want to ensure that *(C1)* each property is associated with a landlord; and *(C2)* landlords cannot apply for benefits related to a property they hold. Intuitively, constraint *(C1)* is violated since *30 Rossetti* is not associated with a landlord; in contrast, constraint *(C2)* is satisfied since *B. Lyndon* (the only landlord) has not claimed any benefits.

The importance of representing integrity constraints in the unified schema has been stressed in many works on data integration (e.g., see [Lenzerini, 2002; Calì *et al.*, 2004]). Faithfully representing such constraints in OBDA is, however, challenging due to the well-known mismatch between the closed-world semantics of integrity constraints in databases and the open-world semantics of ontology languages [Sequeda *et al.*, 2012; Calvanese *et al.*, 2014].

Integrity constraints in relational databases are used to check whether the given data satisfies certain conditions. For instance, one may introduce a functional dependency *(C3)* in Table S stating that no one can claim benefits on two different properties, which holds in our example. During database updates, a check is performed to see whether a person is now claiming for different properties, in which case the update is rejected; if the update is accepted, however, the constraint can be safely disregarded for the purpose of query answering.

The obvious representation of global constraints *(C1)* and *(C3)* using ontology axioms (8) and (9) below, however, yields a markedly different behaviour.

$$\text{Property} \sqsubseteq \exists \text{heldBy} \quad (8) \qquad (\text{func claims}) \quad (9)$$

In particular, adding axiom (8) to the ontology would not raise an error in the data; instead, it would trigger the inference that *30 Rossetti* is associated with some (unknown)

landlord. Furthermore, in contrast to database integrity constraints, axiom (8) affects query answering; e.g., when given a query asking for all properties associated to some landlord, an OBDA system will return *30 Rossetti* as part of the answer even though the owner of this property is not given in the data. Ontology languages used in OBDA are thus closely related to incomplete databases, where data is assumed to be partially specified and dependencies are used to extend the database with missing information.

The problem of combining ontology languages with integrity constraints has received significant attention in the literature [Donini *et al.*, 2002; de Bruijn *et al.*, 2005; Calvanese *et al.*, 2007a; Motik *et al.*, 2009; Motik and Rosati, 2010; Tao *et al.*, 2010; Patel-Schneider and Franconi, 2012; Patel-Schneider, 2015]. In this paper, we explore in the context of OBDA the formalism by Motik *et al.* (2009), where knowledge bases consist of a description logic (DL) TBox and an integrity constraints component where constraints are interpreted as checks over minimal models of the TBox and the data. The framework ensures that, once constraints have been satisfied, they can be safely disregarded for query evaluation. This is a very desirable property in the context of OBDA: constraint satisfaction can be checked whenever the source data is updated, whereas state-of-the-art OBDA systems can continue to be used for query answering without modification.

Our contributions are as follows. In Section 3 we extend the framework by Motik *et al.* (2009) to OBDA. In addition to a TBox $\mathcal{T}$, mappings M, and source database D typically considered in the OBDA literature, we also consider sets of *source constraints* Σ and *target constraints* $\mathcal{C}$. Satisfaction of Σ is defined as usual in databases, by requiring $D \models \Sigma$. In contrast, satisfaction of $\mathcal{C}$ requires that the constraints hold in a suitable minimal model of $\mathcal{T} \cup M \cup D$. We argue that all the desirable properties proved in [Motik *et al.*, 2009] seamlessly extend to OBDA. In our framework, there is also a clear need to understand the semantic relationship between sets of source and target constraints; for instance, when *bootstrapping* a source schema, we may want to check whether the source constraints have been faithfully propagated to the target schema [Sequeda *et al.*, 2012; Calvanese *et al.*, 2014; Console and Lenzerini, 2014; Console, 2016]. To this end, we define the *source-to-target implication problem*, which is to check whether a target constraint γ is satisfied (with respect to $\mathcal{T}$ and M) by all databases D satisfying Σ, in which case γ does not impose additional requirements to the data. Conversely, we define *target-to-source implication*, which is to check whether a source constraint σ is satisfied by all databases D satisfying the target constraints $\mathcal{C}$ with respect to $\mathcal{T}$ and M, in which case σ is captured by $\mathcal{C}$. These problems are strongly related to dependency propagation—a classical problem in databases [Klug, 1980; Klug and Price, 1982; Sagiv, 1987; Abiteboul and Hull, 1988; Fan *et al.*, 2008]; indeed, many of our lower bounds hold for the empty TBox and thus transfer to the database setting.

In Section 4 we study the satisfaction problem for target constraints consisting of disjunctive tuple-generating dependencies and show that it is Π_2^p-complete if the TBox is expressed in *DL-Lite$_\mathcal{R}$*. We also study the data complexity of this problem by assuming that the TBox, mappings, and con-

straints are fixed, and establish tractability in LOGSPACE, which suggests feasibility in practice. To the best of our knowledge, these are the first complexity bounds for constraint satisfaction as defined in [Motik *et al.*, 2009].

In Section 5 we show that, for *DL-Lite$_\mathcal{R}$* TBoxes, source-to-target implication is decidable whenever the language of source constraints Σ has decidable finite entailment problem. Furthermore, we establish tight 2EXPTIME and EXPTIME bounds for Σ consisting of frontier-guarded tuple-generating dependencies [Baget *et al.*, 2011] and for Σ consisting of full dependencies, respectively. We also establish PSPACE-completeness for Σ a set of linear dependencies, and Π_2^p-completeness for Σ a set of functional dependencies.

Finally, in Section 6 we study target-to-source implication and establish complexity bounds for the case where constraints and mappings are frontier-guarded, and $\mathcal{T}$ is a *DL-Lite$_\mathcal{R}$* TBox satisfying certain restrictions. Our upper bounds are obtained via reduction to satisfiability of guarded-negation (GNFO) first-order sentences [Bárány *et al.*, 2015].

2 Preliminaries

We adopt standard notions in first-order logic with equality ($\approx$). Entailment and satisfaction of sentences by interpretations ($\models$) are defined in the standard way. The size $|\varphi|$ of a formula (or set of formulas) φ is also defined as usual.

Dependencies. A *dataset* is a finite set of ground atoms not mentioning equality or function symbols. A *disjunctive tuple-generating dependency* (DTGD) is a function-free sentence σ of the form $\forall \mathbf{xy}. (\varphi(\mathbf{x}, \mathbf{y}) \rightarrow \bigvee_{i=1}^{n} \exists \mathbf{z}_i. (\psi_i(\mathbf{x}, \mathbf{z}_i)))$, where $n \geq 1$, $\mathbf{x}, \mathbf{y}, \mathbf{z}_1, \ldots, \mathbf{z}_n$ are pairwise disjoint tuples of variables, and $\varphi(\mathbf{x}, \mathbf{y}), \psi_1(\mathbf{x}, \mathbf{z}_1), \ldots, \psi_n(\mathbf{x}, \mathbf{z}_n)$ are conjunctions of equality-free atoms over all the given variables and possibly also constants. Formula $\varphi(\mathbf{x}, \mathbf{y})$ is the body of σ, whereas formula $\bigvee_{i=1}^{n} \exists \mathbf{z}_i. \psi_i(\mathbf{x}, \mathbf{z}_i)$ is the head. If $n = 1$, then σ is called a TGD and if, in addition, $\varphi(\mathbf{x}, \mathbf{y})$ comprises at most one atom, then it is called *linear*. A *full dependency* is a TGD without existentially quantified variables and where the head consists of a single atom. An *inclusion dependency* is a linear TGD with a single atom in its head.[1] A *DL-Lite$_\mathcal{R}$ dependency* is an inclusion dependency with predicates of arity one or two and atoms mentioning no constants or repeating variables. A *functional dependency* (FD) on an n-ary predicate P between two sets of positions $\emptyset \subset X, Y \subseteq \{1, \ldots, n\}$ is a function-free sentence of the form $\forall \mathbf{xy}. (P(\mathbf{x}) \wedge P(\mathbf{y}) \rightarrow \bigwedge_{j \in Y} x_j \approx y_j)$, with $\mathbf{x} = x_1, \ldots, x_n$, $\mathbf{y} = y_1, \ldots, y_n$, and $y_i = x_i$ iff $i \in X$. A *dependency* σ is a DTGD or an FD. The *frontier* of σ is the (possibly empty) set of universally quantified variables occurring in both the body and the head. We say that σ is *frontier-guarded* (respectively, *guarded*) if the body has an atom mentioning all frontier variables (respectively, all its variables). The *schema* of a set of dependencies Σ is the set $Sch(\Sigma)$ of predicates in Σ different from $\approx$. From now onwards, universal quantifiers in dependencies are omitted for brevity. We

---

[1] Our definition is more general than the standard one, as it allows for atoms with constants and multiple occurrences of a variable; this facilitates exposition and has no impact on lower bounds.

use Skolemisation to interpret dependencies in Herbrand interpretations. In particular, for each dependency σ with frontier $\mathbf{x}$ and each existentially quantified variable z_i, we define a globally unique function symbol f_σ^i of arity $|\mathbf{x}|$, and we let substitution θ_{sk} map each z_i in σ to its corresponding functional term $f_\sigma^i(\mathbf{x})$. The Skolemisation of σ is the sentence obtained by applying θ_{sk} to σ; the Skolemisation $\mathsf{sk}(\Sigma)$ of a set of dependencies Σ is defined in the obvious way.

Query Answering. A *conjunctive query* (CQ) is a function-free and equality-free first-order formula $q(\mathbf{x})$ constructed using conjunction and existential quantification only. A union of conjunctive queries (UCQ) is a formula $\bigvee_{i=1}^n q_i(\mathbf{x})$, where each $q_i(\mathbf{x})$ is a CQ. The schema of a UCQ q is the set $Sch(q)$ of predicates used in q. A *certain answer* to a UCQ $q(\mathbf{x})$ over dependencies Σ and a dataset D is a tuple $\mathbf{c}$ of constants satisfying $\Sigma \cup D \models q(\mathbf{c})$; the set of such certain answers is denoted as $q^{(\Sigma,D)}$. If Σ consists of TGDs only, $\mathsf{sk}(\Sigma) \cup D$ admits a unique minimal Herbrand model H such that, for each UCQ $q(\mathbf{x})$ and tuple $\mathbf{c}$ of constants, $H \models q(\mathbf{c})$ iff $\mathbf{c} \in q^{(\Sigma,D)}$.

OBDA. We assume countably infinite and pairwise disjoint sets $\mathbf{S}$ and $\mathbf{T}$ of source and target predicates respectively. Dependency σ is a source dependency if it satisfies $Sch(\sigma) \subseteq \mathbf{S}$, and it is a target dependency if $Sch(\sigma) \subseteq \mathbf{T}$. Source and target datasets are defined analogously. As usual in OBDA, we restrict ourselves to *DL-Lite$_\mathcal{R}$* ontologies and GAV mappings. A TBox is a finite set $\mathcal{T}$ of *DL-Lite$_\mathcal{R}$* target dependencies. A *mapping set* M is a finite set of constant-free full dependencies with non-empty frontier where all body predicates belong to $\mathbf{S}$ and all head predicates belong to $\mathbf{T}$. We say that M is *injective* if no two dependencies share a head predicate and that M is *trivial* if it is injective and has only dependencies of the form $P(x_1,\dots,x_n) \to P'(x_1,\dots,x_n)$. The *virtual image* of a mapping set M and a source dataset D is the target dataset $\mathcal{V}_{M,D}$ consisting of all atoms $P(\mathbf{c})$ such that $\varphi(\mathbf{x},\mathbf{y}) \to P(\mathbf{x})$ is in M and $D \models \exists \mathbf{y}.\, \varphi(\mathbf{c},\mathbf{y})$. The certain answers $q^{(\mathcal{T} \cup M, D)}$ to a query q over the target vocabulary coincide with $q^{(\mathcal{T}, \mathcal{V}_{M,D})}$. The *unfolding* of a formula φ with respect to M, where $Sch(\varphi) \subseteq \mathbf{T}$, is the formula $\mathsf{unfold}_M(\varphi)$ obtained from φ by replacing each atom $P(\mathbf{t})$ with $\bot$ (if M does not mention P) or with $\bigvee_{i=1}^n \exists \mathbf{y}_i.\, \varphi_i(\mathbf{t}, \mathbf{y}_i)$, where $\varphi_1(\mathbf{x}_1, \mathbf{y}_1) \to P(\mathbf{x}_1), \dots, \varphi_n(\mathbf{x}_n, \mathbf{y}_n) \to P(\mathbf{x}_n)$ are all mappings in M for P. The unfolding of a set of formulas is defined in the obvious way.

GNFO. We also consider the *guarded-negation fragment of first-order logic* (GNFO) [Bárány *et al.*, 2015] consisting of formulas φ specified by the following grammar, where α ranges over atoms, and expressions $\alpha(\mathbf{x}, \mathbf{y})$ and $\varphi(\mathbf{x})$ denote that $\mathbf{x} \cup \mathbf{y}$ and $\mathbf{x}$ are the free variables in α and φ, respectively:

$$\varphi := \alpha \mid \exists x.\, \varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \alpha(\mathbf{x}, \mathbf{y}) \wedge \neg \varphi(\mathbf{x}).$$

The restriction over first-order logic is the requirement that all free variables of a negated expression must be guarded by an atom. GNFO enjoys the finite model property, and satisfiability checking is 2EXPTIME-complete [Bárány *et al.*, 2015].

3 OBDA with Integrity Constraints

We next present our extension of OBDA with integrity constraints, and define the relevant reasoning problems. In addi-

tion to a TBox $\mathcal{T}$, mappings M, and source dataset D, we also consider finite sets of source dependencies Σ (the *source constraints*) and target dependencies $\mathcal{C}$ (the *target constraints*). Satisfaction of Σ is defined as usual in databases, by assuming that source data is complete and requiring $D \models \Sigma$. In contrast, satisfaction of target constraints $\mathcal{C}$ must be consistent with the open-world semantics of OBDA, where query answering is defined in terms of *all* models of $\mathcal{T} \cup \mathcal{V}_{M,D}$.

Motik *et al.* (2009) proposed a formalism that combines a DL ontology consisting of TBox $\mathcal{T}$ and target dataset A with integrity constraints $\mathcal{C}$. Constraints $\mathcal{C}$ are satisfied if they hold in all minimal Herbrand models of $\mathsf{sk}(\mathcal{T}) \cup A$, in which case they do not influence query answering. Skolemisation is introduced to address the semantic difficulties posed by existential quantifiers in the TBox: it ensures that equivalent TBoxes satisfy the same constraints. The following definition extends the semantics by Motik *et al.* to OBDA in the obvious way.

Definition 1. *A source dataset D satisfies a set $\mathcal{C}$ of target constraints with respect to a TBox $\mathcal{T}$ and a mapping set M, written $D \models_{[M,\mathcal{T}]} \mathcal{C}$, if the minimal Herbrand model of $\mathsf{sk}(\mathcal{T}) \cup \mathcal{V}_{M,D}$ is also a model of $\mathcal{C}$.*

Example 2. *Let D, M, and $\mathcal{T}$ be as in Example 1, and let $\mathcal{C} = \{\gamma_1\}$, where γ_1 formalises constraint (C1) as the dependency $\mathsf{Property}(x) \to \exists z.\, \mathsf{heldBy}(x,z) \wedge \mathsf{Landlord}(z)$. Then, as intuitively expected, $D \not\models_{[M,\mathcal{T}]} \mathcal{C}$ since the minimal Herbrand model of $\mathsf{sk}(\mathcal{T}) \cup \mathcal{V}_{M,D}$ satisfies $\mathsf{Property}(\text{30 Rossetti})$ and does not satisfy $\exists z.\, \mathsf{heldBy}(\text{30 Rossetti}, z)$.*

As in [Motik *et al.*, 2009], the definition of constraint satisfaction ensures that target constraints satisfied by the data can be safely disregarded for query answering.

Theorem 1. *If $D \models_{[M,\mathcal{T}]} \mathcal{C}$, then $q^{(\mathcal{T} \cup M \cup \mathcal{C}, D)} = q^{(\mathcal{T} \cup M, D)}$ for any UCQ q with $Sch(q) \subseteq \mathbf{T}$.*

In contrast to Motik *et al.* (2009), our OBDA framework defines two types of constraints, which are satisfied according to different semantics. In this setting, there is a clear need to analyse the semantic relationship between source and target constraints, e.g., to understand whether a source constraint is propagated via the mappings to the target. To this end, we next formalise the notion of *constraint implication*.

Definition 2. *Let Σ and $\mathcal{C}$ be sets of source and target constraints respectively, let $\mathcal{T}$ be a TBox and M a mapping set.*

We say that Σ implies a target constraint γ with respect to $\mathcal{T}$ and M, written $\Sigma \models_{[M,\mathcal{T}]} \gamma$, if, for every source dataset D, it holds that $D \models_{[M,\mathcal{T}]} \gamma$ whenever $D \models \Sigma$.

Conversely, $\mathcal{C}$ implies a source constraint σ with respect to $\mathcal{T}$ and M, written $\mathcal{C} \models_{[M,\mathcal{T}]} \sigma$, if, for every source dataset D, it holds that $D \models \sigma$ whenever $D \models_{[M,\mathcal{T}]} \mathcal{C}$.

Example 3. *Let D, M, and $\mathcal{T}$ be as in Example 1, and let $\Sigma = \{\sigma_1\}$ where σ_1 formalises constraint (C3) in the introduction as $\mathsf{S}(x,x_1) \wedge \mathsf{S}(x,x_2) \to x_1 \approx x_2$. For the target constraint γ_1 from Example 2, we have that $D \not\models_{[M,\mathcal{T}]} \gamma_1$ and because $D \models \Sigma$, we derive that $\Sigma \not\models_{[M,\mathcal{T}]} \gamma_1$. However, if we extend Σ with the inclusion dependency $\sigma_2 = \mathsf{S}(y,x) \to \exists z_1, z_2.\, \mathsf{T}(x, z_1, z_2)$, we can then verify that $\Sigma \models_{[M,\mathcal{T}]} \gamma_1$.*

Let $\gamma_2 = \text{claims}(x, x_1) \wedge \text{claims}(x, x_2) \rightarrow x_1 \approx x_2$ and $\gamma_3 = \text{Claimant}(x) \rightarrow \exists z_1, z_2. \text{claims}(x, z_1) \wedge \text{heldBy}(z_1, z_2)$. For $\mathcal{C} = \{\gamma_2, \gamma_3\}$ one can verify that $\mathcal{C} \models_{[M,\mathcal{T}]} \sigma_1$ as a result of γ_2 being in $\mathcal{C}$, and also that $\mathcal{C} \models_{[M,\mathcal{T}]} \sigma_2$.

4 Constraint Satisfaction

In this section we study the complexity of checking $D \models_{[M,\mathcal{T}]} \mathcal{C}$ for $\mathcal{T}$ a *DL-Lite$_{\mathcal{R}}$* TBox.

Motik *et al.* (2009) proved decidability of constraint satisfaction for $\mathcal{T}$ an $\mathcal{ALCHI}$ TBox and $\mathcal{C}$ consisting of arbitrary first-order sentences. Although decidability extends to our setting, the proof in [Motik *et al.*, 2009] is by reduction to satisfiability of MSO formulae on infinite k-ary trees (SkS) and thus only provides a non-elementary upper bound.

We next show that the problem becomes Π_2^p-complete if $\mathcal{T}$ is *DL-Lite$_{\mathcal{R}}$* and $\mathcal{C}$ consists of DTGDs. The lower bound follows from a result by Pichler and Skritek (2011), who showed that the problem of checking whether a dataset satisfies a TGD is Π_2^p-hard. To establish the upper bound we rely on [Calì *et al.*, 2012] and show that, if a target constraint γ does not hold in the minimal Herbrand model H of $\text{sk}(\mathcal{T}) \cup \mathcal{V}_{M,D}$ (and hence $D \not\models_{[M,\mathcal{T}]} \gamma$), then γ is violated already in a fragment of H where the depth of all functional terms is bounded by a polynomial in the size of $\mathcal{T}$, γ, and M.

Definition 3. *Let $\mathcal{T}$ be a DL-Lite$_{\mathcal{R}}$ TBox, let A be a target dataset, and let $\gamma = \varphi(\mathbf{x}, \mathbf{y}) \rightarrow \bigvee_{i=1}^{n} \exists \mathbf{z}_i. \psi_i(\mathbf{x}, \mathbf{z}_i)$ be a target constraint. A* counter-example *to γ with respect to $\mathcal{T} \cup A$ is a mapping λ of $\mathbf{x}$ to terms in the Herbrand Universe of $\text{sk}(\mathcal{T}) \cup A$ such that $\text{sk}(\mathcal{T}) \cup A \models \exists \mathbf{y}.\varphi(\lambda(\mathbf{x}), \mathbf{y})$ but $\text{sk}(\mathcal{T}) \cup A \not\models \bigvee_{i=1}^{n} \exists \mathbf{z}_i. \psi_i(\lambda(\mathbf{x}), \mathbf{z}_i)$.*

Lemma 1. *Assume that $D \not\models_{[M,\mathcal{T}]} \gamma$, for γ a target constraint of the form $\varphi(\mathbf{x}, \mathbf{y}) \rightarrow \bigvee_{i=1}^{n} \exists \mathbf{z}_i. \psi_i(\mathbf{x}, \mathbf{z}_i)$. Let H be the minimal Herbrand model of $\mathcal{T} \cup \mathcal{V}_{M,D}$ and let H_d be the subset of H involving terms of functional depth at most $d = 16 \cdot p \cdot |\gamma|$, with p the number of target predicates in $\mathcal{T}$, M, and γ. Then, there exists a counter-example λ to γ with respect to $\mathcal{T} \cup \mathcal{V}_{M,D}$ such that $H_d \models \exists \mathbf{y}. \varphi(\lambda(\mathbf{x}), \mathbf{y})$.*

The result in the lemma immediately suggests a non-deterministic algorithm for checking the complement of constraint satisfaction. To check that $D \not\models_{[M,\mathcal{T}]} \gamma$, we

- guess a mapping λ from the frontier variables $\mathbf{x}$ of γ to ground terms of functional depth at most d involving constants from D; and
- check that $\text{sk}(\mathcal{T}) \cup \mathcal{V}_{M,D} \models \exists \mathbf{y}. \varphi(\lambda(\mathbf{x}), \mathbf{y})$ and $\text{sk}(\mathcal{T}) \cup \mathcal{V}_{M,D} \not\models \bigvee_{i=1}^{n} \exists \mathbf{z}_i. \psi_i(\lambda(\mathbf{x}), \mathbf{z}_i)$ using two NP oracles.

Theorem 2. *Checking $D \models_{[M,\mathcal{T}]} \mathcal{C}$ is Π_2^p-complete if $\mathcal{T}$ is DL-Lite$_{\mathcal{R}}$ and $\mathcal{C}$ consists of DTGDs; the lower bound holds already if we require $\mathcal{T} = \emptyset$, and $\mathcal{C}$ to consist of a single TGD.*

We conclude this section by showing that constraint satisfaction is in LOGSPACE in data complexity under the assumptions of Theorem 2. The result also relies on Lemma 1, where the depth d is now constant since $\mathcal{T}$, M, and $\mathcal{C}$ are fixed. Furthermore, $\mathcal{V}_{M,D}$ and the relevant model fragment H_d can be generated using logarithmic space. Finally, checking $H_d \models \gamma$ is clearly feasible in LOGSPACE for fixed γ.

Theorem 3. *For $\mathcal{T}$ a fixed DL-Lite$_{\mathcal{R}}$ TBox, $\mathcal{C}$ a fixed set of DTGDs, and M a fixed mapping set, checking $D \models_{[M,\mathcal{T}]} \mathcal{C}$ for an input source dataset D is in LOGSPACE.*

5 Source-to-Target Constraint Implication

We now turn our attention to the problem of checking whether $\Sigma \models_{[M,\mathcal{T}]} \gamma$ for $\mathcal{T}$ a *DL-Lite$_{\mathcal{R}}$* TBox.

We show decidability for any class of source constraints with decidable finite entailment problem. Furthermore, we establish tight complexity bounds showing that source-to-target implication is no harder than finite entailment for the following classes of source dependencies: *(i)* frontier-guarded TGDs, *(ii)* full dependencies, and *(iii)* linear dependencies. We also show Π_2^p-completeness for Σ consisting of FDs. All our lower bounds hold for $\mathcal{T} = \emptyset$, which allows us to obtain new results for the dependency propagation problem in databases. Finally, we show that considering TBoxes based on full dependencies immediately leads to undecidability.

We next describe an algorithm for checking $\Sigma \not\models_{[M,\mathcal{T}]} \gamma$ when Σ is an arbitrary set of dependencies, $\mathcal{T}$ is a *DL-Lite$_{\mathcal{R}}$* TBox, and γ is a DTGD of the form $\varphi(\mathbf{x}, \mathbf{y}) \rightarrow \bigvee_{i=1}^{n} q_i(\mathbf{x})$ with each q_i a CQ. Our algorithm searches for a witness source dataset D_w satisfying Σ but not γ with respect to $\mathcal{T}$ and M. In doing so, it relies on two main observations.

First, each counter-example λ to γ w.r.t. $\mathcal{T} \cup \mathcal{V}_{M,D_w}$ is justified by a subset D of D_w containing linearly many constants and atoms in the size of γ and M. Our algorithm will check $D \models_{[M,\mathcal{T}]} \gamma$ for all D up to the relevant size; as discussed in Section 4, this check can be done by considering a shallow fragment of the minimal Herbrand model H of $\text{sk}(\mathcal{T}) \cup \mathcal{V}_{M,D}$.

Second, any such source dataset D with $D \not\models_{[M,\mathcal{T}]} \gamma$ as per counter-example λ may not satisfy Σ, in which case we need to extend D to a (finite) model D' of Σ. Such extension, however, may also yield an extended virtual image $\mathcal{V}_{M,D'}$ which may invalidate λ if $\text{sk}(\mathcal{T}) \cup \mathcal{V}_{M,D'} \models q_i(\lambda(\mathbf{x}))$ for some CQ q_i in the head of γ. Thus, our algorithm checks whether D can be extended to a dataset D' satisfying Σ and preserving the counter-example λ (and hence still violating γ). To this end, we next introduce the notion of a λ-witness.

Definition 4. *Let $\mathcal{T}$ be a DL-Lite$_{\mathcal{R}}$ TBox, let $q(\mathbf{x})$ be a CQ over $\mathbf{T}$, and let λ map $\mathbf{x}$ to ground terms. A λ-witness for $\mathcal{T}$ and $q(\mathbf{x})$ is a Boolean CQ $q' = \exists \mathbf{y}. \xi(\mathbf{y})$ over predicates in $\mathbf{T}$ such that $\text{sk}(\mathcal{T}) \cup \xi(\lambda'(\mathbf{y})) \models q(\lambda(\mathbf{x}))$ for some injective mapping λ' from $\mathbf{y}$ to constants.*

Intuitively, a λ-witness for $\mathcal{T}$ and a CQ $q_i(\mathbf{x})$ in the head of γ specifies a pattern in the target data, the presence of which is sufficient to invalidate λ. Thus, to ensure that a candidate D not satisfying γ as per λ can be extended to a (finite) model of Σ that preserves λ, it suffices to show that $\Sigma \cup D$ does not finitely entail the unfolding of any λ-witness. Finally, since $\mathcal{T}$ is *DL-Lite$_{\mathcal{R}}$*, the size of the relevant λ-witnesses to consider can be bounded by the size of γ.

With these ingredients at hand, we are ready to formalise our algorithm. Let m_v and m_a be the maximum number of variables and atoms in a mapping from M, and $\gamma = \varphi(\mathbf{x}, \mathbf{y}) \rightarrow \bigvee_{i=1}^{n} q_i(\mathbf{x})$. We then apply the following steps.

1) Do the following for each source dataset D with at most $m_a \cdot |\gamma|$ atoms defined over the predicates from M and the constants in γ extended with $m_v \cdot |\gamma|$ fresh constants.

   a) Compute the subset H_d of the minimal Herbrand model H of $\mathsf{sk}(\mathcal{T}) \cup \mathcal{V}_{M,D}$ involving terms of functional depth at most d, where d is the quantity defined in Lemma 1 for M, $\mathcal{T}$, and γ.

   b) For each counter-example λ to γ w.r.t. $\mathcal{T} \cup \mathcal{V}_{M,D}$ involving only terms in H_d do the following.

      i) Compute the set W of all λ-witnesses q' for $\mathcal{T}$ and a CQ $q_i(\mathbf{x})$ in γ such that $|q'| \leq |q_i|$ and q' uses only target predicates from M, constants from $q_i(\lambda(\mathbf{x}))$, and quantified variables $y_1, \ldots, y_{2 \cdot |q_i|}$.

      ii) Compute $Q_\lambda = \mathsf{unfold}_M(\bigvee_{q' \in W} q')$.

      iii) If $\Sigma \cup D$ does not finitely entail Q_λ, return `true`.

2) Return `false`.

Correctness follows from our previous observations. If the algorithm returns true, then it has found a source dataset D and counter-example λ to γ such that D can be extended to a model D' of Σ not satisfying Q_λ; the construction of Q_λ and the properties of unfolding ensure that $\mathcal{V}_{M,D'}$ does not contain a pattern that can make some CQ $q_i(\lambda(\mathbf{x}))$ true in $\mathsf{sk}(\mathcal{T}) \cup \mathcal{V}_{M,D'}$, thus, λ remains a counter-example. Conversely, $\Sigma \not\models_{[M,\mathcal{T}]} \gamma$ is justified by a D and a counter-example λ involving functional terms of polynomial depth; our algorithm will consider (an isomorphism of) λ and determine existence of a dataset satisfying Σ and preserving λ. The following decidability result immediately follows.

Theorem 4. *Checking* $\Sigma \models_{[M,\mathcal{T}]} \gamma$ *is decidable if* $\mathcal{T}$ *is DL-Lite$_\mathcal{R}$,* γ *is a DTGD, and* Σ *is in a class of dependencies for which UCQ entailment w.r.t. finite models is decidable.*

Our algorithm considers exponentially many source datasets of polynomial size; for each of them, the relevant part of the corresponding minimal Herbrand model can be constructed in exponential time. Furthermore, there are exponentially many counter-examples λ to consider and also exponentially many λ-witnesses. Thus, the algorithm performs exponentially many finite query entailment tests. This shows that source-to-target implication is no harder than finite query entailment if we consider classes of source dependencies for which the latter problem is ExpTime-hard. Thus, our analysis provides 2ExpTime (resp., ExpTime) upper bounds for Σ consisting of frontier-guarded TGDs (resp., full dependencies). Furthermore, by considering a simple variant of our algorithm which avoids the explicit construction of H_d in Step 1a and of W in Step 1(b)i, we can establish a PSPACE upper bound for Σ consisting of linear dependencies. The matching lower bounds are obtained via a reduction from finite entailment, and hold for $\mathcal{T} = \emptyset$ and trivial mappings.

Theorem 5. *Checking* $\Sigma \models_{[M,\mathcal{T}]} \gamma$ *for* $\mathcal{T}$ *a DL-Lite$_\mathcal{R}$ TBox and* γ *a DTGD is:*
1) 2ExpTime-complete for Σ *frontier-guarded TGDs;*
2) ExpTime-complete for Σ *full dependencies; and*
3) PSPACE-complete for Σ *linear dependencies.*
The lower bounds hold already for M *trivial and* $\mathcal{T} = \emptyset$.

We next show that source-to-target implication becomes Π_2^p-complete for Σ consisting of functional dependencies. The lower bound follows by a reduction from CQ containment w.r.t. a *DL-Lite$_\mathcal{R}$* TBox [Bienvenu *et al.*, 2012]. The key observation to establish the upper bound is that it suffices to consider source datasets D of polynomial size. Hence, to decide the complement of our problem, we can guess D and accept if $D \models \Sigma$ and $D \not\models_{[M,\mathcal{T}]} \gamma$. The former can be checked in polynomial time; the latter can be checked by applying the algorithm in Section 4, which involves additional (existential) guesses and two NP oracle calls.

Theorem 6. *Checking* $\Sigma \models_{[M,\mathcal{T}]} \gamma$ *is* Π_2^p-*complete if* Σ *consists of FDs,* $\mathcal{T}$ *is DL-Lite$_\mathcal{R}$, and* γ *is a DTGD. The lower bound holds already for* M *a trivial mapping set and* $\Sigma = \emptyset$.

We conclude this section by showing that extending the ontology language can easily lead to undecidability. In particular, if $\mathcal{T}$ is Datalog (i.e., consists of full dependencies) then we can easily establish a reduction from the non-uniform containment problem for Datalog programs [Shmueli, 1993].

Theorem 7. *Checking* $\Sigma \models_{[M,\mathcal{T}]} \gamma$ *is undecidable if* $\mathcal{T}$ *and* γ *are full dependencies, and even if* $\Sigma = \emptyset$ *and* M *is trivial.*

6 Target-to-Source Constraint Implication

We finally study the implication problem of checking whether $\mathcal{C} \models_{[M,\mathcal{T}]} \sigma$. We show decidability for the case where σ, M, and $\mathcal{C}$ are frontier-guarded, and $\mathcal{T}$ consists of full *DL-Lite$_\mathcal{R}$* dependencies; we leave for future work the investigation of the problem for $\mathcal{T}$ an arbitrary *DL-Lite$_\mathcal{R}$* TBox.

To check $\mathcal{C} \models_{[M,\mathcal{T}]} \sigma$ we proceed as follows:

1) We rewrite $\mathcal{C}$ with respect to $\mathcal{T}$ to obtain a new set $\mathcal{C}'$ of target constraints. This allows us to eliminate $\mathcal{T}$ since we can show that $\mathcal{C} \models_{[M,\mathcal{T}]} \sigma$ if and only if $\mathcal{C}' \models_{[M,\emptyset]} \sigma$.

2) We unfold $\mathcal{C}'$ with respect to the mappings M.

3) We check finite satisfiability of $\mathsf{unfold}_M(\mathcal{C}') \cup \{\neg\sigma\}$.

In Step 1, we rewrite each DTGD $\gamma = \varphi(\mathbf{x}, \mathbf{y}) \rightarrow \bigvee_{i=1}^n q_i(\mathbf{x})$ with respect to $\mathcal{T}$ using standard rewriting techniques [Calvanese *et al.*, 2007b]; in particular, we compute a UCQ rewriting ρ_2 of the head UCQ $\bigvee_{i=1}^n q_i(\mathbf{x})$ and a UCQ rewriting ρ_1 of the body query $\exists \mathbf{y}.\varphi(\mathbf{x}, \mathbf{y})$ to obtain the set of DTGDs of the from $q \rightarrow \rho_2$, with q a CQ in ρ_1.

Definition 5. *Let* $q(\mathbf{x})$ *be a UCQ and let* $\mathcal{T}$ *be a DL-Lite$_\mathcal{R}$ TBox. A UCQ* $\rho(\mathbf{x})$ *is a* $\mathcal{T}$-*rewriting of* q *if, for every target dataset* A, *it holds that* $q^{(\mathcal{T},A)} = \rho^{(\emptyset,A)}$.

Lemma 2. *Let* $\mathcal{T}$ *consist of full DL-Lite$_\mathcal{R}$ dependencies and let* $\mathcal{C}$ *consist of DTGDs. For each* $\gamma \in \mathcal{C}$, *let* ρ_1^γ *and* ρ_2^γ *be* $\mathcal{T}$-*rewritings of the body and head of* γ *respectively, let* $\mathcal{C}'_\gamma = \{q \rightarrow \rho_2^\gamma \mid q$ *a CQ in* $\rho_1^\gamma\}$, *and let* $\mathcal{C}' = \bigcup_{\gamma \in \mathcal{C}} \mathcal{C}'_\gamma$.
Then, for each source dataset D, *it holds that* $D \models_{[M,\mathcal{T}]} \mathcal{C}$ *if and only if* $D \models_{[M,\emptyset]} \mathcal{C}'$.

Lemma 2 immediately implies that $\mathcal{C} \models_{[M,\mathcal{T}]} \sigma$ if and only if $\mathcal{C}' \models_{[M,\emptyset]} \sigma$, and hence we have reduced our problem to the particular case where the TBox is empty. As shown in the following example, however, Lemma 2 may not hold if $\mathcal{T}$ is a *DL-Lite$_\mathcal{R}$* TBox with existentially quantified dependencies.

Example 4. *Consider mapping set $M = \{A(x) \rightarrow B(x)\}$, TBox $\mathcal{T} = \{B(x) \rightarrow \exists y.\, R(x,y)\}$, and target constraint $\mathcal{C} = \{R(y,x) \rightarrow F(x)\}$. One can check that $\mathcal{C}' = \mathcal{C}$, where $\mathcal{C}'$ is the rewritten constraints as in Lemma 2. Consider $D = \{A(c)\}$. Clearly, $D \models_{[M,\emptyset]} \mathcal{C}'$; however, the minimal Herbrand model of $\mathsf{sk}(\mathcal{T}) \cup \mathcal{V}_{M,D}$ consists of atoms $B(c), R(c, f(c))$, and hence $D \not\models_{[M,\mathcal{T}]} \mathcal{C}$.*

Steps 2 and 3 are then justified by the following lemma.

Lemma 3. *For σ a DTGD and $\mathcal{C}$ consisting of DTGDs, we have that $\mathcal{C} \models_{[M,\emptyset]} \sigma$ if and only if $\mathsf{unfold}_M(\mathcal{C}) \cup \{\neg\sigma\}$ does not admit a finite model.*

Correctness of our approach follows from Lemmas 2 and 3. Analysis of Steps 1–3 also provides a 3EXPTIME upper bound. On the one hand, a rewritten set of constraints $\mathcal{C}'$ of exponential size satisfying Lemma 2 can be computed in exponential time using standard techniques [Calvanese *et al.*, 2007b]; furthermore, if $\mathcal{C}$ is frontier-guarded we can ensure that so is $\mathcal{C}'$. On the other hand, if $\mathcal{C}'$, M, and σ are frontier-guarded, then $\mathsf{unfold}_M(\mathcal{C}') \cup \{\neg\sigma\}$ can be written as a GNFO formula φ; finite satisfiability can be checked in 2EXPTIME in $|\varphi|$, which in turn is exponential in $\mathcal{T}$, $\mathcal{C}$, and M.

Theorem 8. *Checking $\mathcal{C} \models_{[M,\mathcal{T}]} \sigma$ is in 3EXPTIME for σ a frontier-guarded DTGD, M a frontier-guarded mapping set, $\mathcal{T}$ consisting of full DL-Lite$_\mathcal{R}$ dependencies, and $\mathcal{C}$ consisting of frontier-guarded DTGDs. The problem is 2EXPTIME-hard even if $\mathcal{T} = \emptyset$ and one of the following holds: either (i) σ is an inclusion dependency, M is trivial, and $\mathcal{C}$ is a set of guarded TGDs; or (ii) σ is a linear dependency, M is linear, and $\mathcal{C}$ consists of full and linear dependencies.*

Theorem 8 provides only 2EXPTIME lower bounds via reductions from query answering under guarded (D)TGDs [Calì *et al.*, 2013; Bourhis *et al.*, 2016]. However, we obtain a 2EXPTIME upper bound if we can ensure the set $\mathcal{C}'$ of rewritten target constraints computed in Step 1 is of polynomial size. This is so if $\mathcal{C}$ consists of inclusion dependencies: CQs with a single atom admit polysize $\mathcal{T}$-rewritings whenever $\mathcal{T}$ consists of linear dependencies. The matching 2EXPTIME lower bound is provided by case (ii) in Theorem 8.

Theorem 9. *Checking $\mathcal{C} \models_{[M,\mathcal{T}]} \sigma$ is 2EXPTIME-complete for σ a frontier-guarded DTGD, M consisting of frontier-guarded mappings, $\mathcal{T}$ consisting of full DL-Lite$_\mathcal{R}$ dependencies, and $\mathcal{C}$ consisting of inclusion dependencies.*

We conclude by showing that allowing for arbitrary full dependencies in $\mathcal{C}$ immediately leads to undecidability, even for the empty TBox. This follows by a straightforward reduction from the fact entailment problem for arbitrary TGDs.

Theorem 10. *Checking $\mathcal{C} \models_{[M,\mathcal{T}]} \sigma$ is undecidable if $\mathcal{C}$ and σ are full dependencies, and even if $\mathcal{T} = \emptyset$ and M is injective.*

7 Related Work

Although there have been many proposals for combining ontologies with integrity constraints, there is no consensus around any specific semantics. As discussed in Section 3, we adopted the one in [Motik *et al.*, 2009] as it provides a number of nice theoretical guarantees; it would be interesting to

study OBDA constraint satisfaction and implication under alternative semantics, and the work by Console (2016) could be an interesting starting point in this direction.

Source-to-target constraint implication in the absence of a TBox has been widely studied in the database literature under different names, such as *view dependency inference* and *dependency propagation*. The problem is EXPTIME-complete for full dependencies [Fan, 2009], and we showed in Section 5 that the EXPTIME upper bound extends to *DL-Lite$_\mathcal{R}$* TBoxes and target DTGDs. The problem is NP-complete for FDs and join dependencies [Klug and Price, 1982; Fan, 2009] and becomes tractable if the sets of constraints consist of FDs and multi-valued dependencies [Fan, 2009]. Undecidability for the case where mappings are allowed to contain arbitrary algebra expressions in the body and all dependencies are FDs was shown by Klug (1980). Finally, Abiteboul and Hull (1988) showed decidability for FDs and "TBoxes" consisting of non-recursive full dependencies, where undecidability is obtained if recursion is allowed. Source-to-target implication for a generalisation of FDs has been recently studied in the context of data quality [Fan *et al.*, 2008].

Console and Lenzerini (2014) study the *protection* problem, which is to check whether the virtual image of every source dataset satisfying the given source constraints can be extended to a model of a given *DL-Lite$_\mathcal{A}$* TBox. This problem is closely related to source-to-target implication since, under the Unique Name Assumption, disjointness and functionality axioms in the TBox behave like integrity constraints. They also study the analogue problem of target-to-source implication, called *faithfulness*. Console and Lenzerini establish EXPTIME upper bounds for both problems when source dependencies are weakly-acyclic sets of TGDs and equality-generating dependencies.

Finally, the problem of bootstrapping source schemas into an ontology has been studied in [Sequeda *et al.*, 2012; Calvanese *et al.*, 2014], where certain transformations are proposed for propagating the database constraints to the ontology with the aim of preserving their original semantics.

8 Conclusions and Future Work

We have proposed an extension of OBDA with source and target integrity constraints and studied the relevant reasoning problems. Our complexity results for constraint satisfaction and source-to-target implication are applicable to the ontology and mapping languages typically used in OBDA, and cover a wide range of constraint languages. Our results on target-to-source implication, however, leave a few gaps that we plan to close in future work; in particular, we will extend our results to arbitrary *DL-Lite$_\mathcal{R}$* TBoxes and investigate whether our 3EXPTIME upper bound in Theorem 8 is tight. Finally, it would be interesting to study constraint satisfaction and implication for the alternative semantics of integrity constraints in ontologies mentioned in the related work section.

References

[Abiteboul and Hull, 1988] Serge Abiteboul and Richard Hull. Data functions, datalog and negation (extended abstract). In *SIGMOD'88*, pages 143–153, 1988.

[Baget *et al.*, 2011] Jean-François Baget, Marie-Laure Mugnier, Sebastian Rudolph, and Michaël Thomazo. Walking the complexity lines for generalized guarded existential rules. In *IJCAI'11*, pages 712–717, 2011.

[Bárány *et al.*, 2015] Vince Bárány, Balder ten Cate, and Luc Segoufin. Guarded negation. *J. ACM*, 62(3), 2015.

[Bienvenu *et al.*, 2012] Meghyn Bienvenu, Carsten Lutz, and Frank Wolter. Query containment in description logics reconsidered. In *KR'12*, 2012.

[Bourhis *et al.*, 2016] Pierre Bourhis, Marco Manna, Michael Morak, and Andreas Pieris. Guarded-based disjunctive tuple-generating dependencies. *ACM Trans. Database Syst.*, 41(4):27:1–27:45, 2016.

[Calì *et al.*, 2004] Andrea Calì, Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. Data integration under integrity constraints. *Inf. Syst.*, 29(2):147–163, 2004.

[Calì *et al.*, 2012] Andrea Calì, Georg Gottlob, and Thomas Lukasiewicz. A general datalog-based framework for tractable query answering over ontologies. *JWS*, 2012.

[Calì *et al.*, 2013] Andrea Calì, Georg Gottlob, and Michael Kifer. Taming the infinite chase: Query answering under expressive relational constraints. *JAIR*, 48:115–174, 2013.

[Calvanese *et al.*, 2007a] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. EQL-Lite: Effective first-order query processing in description logics. In *IJCAI'07*, 2007.

[Calvanese *et al.*, 2007b] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *J. Autom. Reasoning*, 39(3):385–429, 2007.

[Calvanese *et al.*, 2014] Diego Calvanese, Wolfgang Fischl, Reinhard Pichler, Emanuel Sallinger, and Mantas Simkus. Capturing relational schemas and functional dependencies in RDFS. In *AAAI'11*, pages 1003–1011, 2014.

[Calvanese *et al.*, 2017] Diego Calvanese, Benjamin Cogrel, Sarah Komla-Ebri, Roman Kontchakov, Davide Lanti, Martin Rezk, Mariano Rodriguez-Muro, and Guohui Xiao. Ontop: Answering SPARQL queries over relational databases. *Semantic Web*, 8(3):471–487, 2017.

[Console and Lenzerini, 2014] Marco Console and Maurizio Lenzerini. Data quality in ontology-based data access: The case of consistency. In *AAAI'14*, pages 1020–1026, 2014.

[Console, 2016] Marco Console. Data accuracy as knowledge in ontology based data access (preliminary report). In *Proceedings of the 29th International Workshop on Description Logics*, 2016.

[de Bruijn *et al.*, 2005] Jos de Bruijn, Rubén Lara, Axel Polleres, and Dieter Fensel. OWL DL vs. OWL flight: conceptual modeling and reasoning for the semantic web. In *WWW'05*, pages 623–632, 2005.

[Donini *et al.*, 2002] Francesco M. Donini, Daniele Nardi, and Riccardo Rosati. Description logics of minimal knowledge and negation as failure. *TOCL*, 3(2), 2002.

[Fan *et al.*, 2008] Wenfei Fan, Shuai Ma, Yanli Hu, Jie Liu, and Yinghui Wu. Propagating functional dependencies with conditions. *PVLDB*, 1(1):391–407, 2008.

[Fan, 2009] Wenfei Fan. Implication of constraints. In Ling Liu and M. Tamer Özsu, editors, *Encyclopedia of Database Systems*, pages 1396–1400. Springer, 2009.

[Kharlamov *et al.*, 2017] Evgeny Kharlamov, Dag Hovland, Martin G. Skjæveland, Dimitris Bilidas, Ernesto Jiménez-Ruiz, Guohui Xiao, Ahmet Soylu, Davide Lanti, Martin Rezk, Dmitriy Zheleznyakov, Martin Giese, Hallstein Lie, Yannis E. Ioannidis, Yannis Kotidis, Manolis Koubarakis, and Arild Waaler. Ontology based data access in Statoil. *J. Web Sem.*, 44:3–36, 2017.

[Klug and Price, 1982] Anthony C. Klug and Rod Price. Determining view dependencies using tableaux. *ACM Trans. Database Syst.*, 7(3):361–380, 1982.

[Klug, 1980] Anthony C. Klug. Calculating constraints on relational expressions. *TODS*, 5(3):260–290, 1980.

[Lenzerini, 2002] Maurizio Lenzerini. Data integration: A theoretical perspective. In *PODS'02*, 2002.

[Motik and Rosati, 2010] Boris Motik and Riccardo Rosati. Reconciling description logics and rules. *J. ACM*, 57(5):30:1–30:62, 2010.

[Motik *et al.*, 2009] Boris Motik, Ian Horrocks, and Ulrike Sattler. Bridging the gap between OWL and relational databases. *J. Web Sem.*, 7(2):74–89, 2009.

[Patel-Schneider and Franconi, 2012] Peter F. Patel-Schneider and Enrico Franconi. Ontology constraints in incomplete and complete data. In *ISWC'12*, 2012.

[Patel-Schneider, 2015] Peter F. Patel-Schneider. Using description logics for RDF constraint checking and closed-world recognition. In *AAAI'15*, pages 247–253, 2015.

[Pichler and Skritek, 2011] Reinhard Pichler and Sebastian Skritek. The complexity of evaluating tuple generating dependencies. In *ICDT'11*, pages 244–255, 2011.

[Poggi *et al.*, 2008] Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. Linking data to ontologies. *Journal on Data Semantics*, 10:133–173, 2008.

[Sagiv, 1987] Yehoshua Sagiv. Optimizing datalog programs. In *PODS'87*, pages 349–362, 1987.

[Sequeda *et al.*, 2012] Juan F. Sequeda, Marcelo Arenas, and Daniel P. Miranker. On directly mapping relational databases to RDF and OWL. In *WWW'12*, 2012.

[Shmueli, 1993] Oded Shmueli. Equivalence of DATALOG queries is undecidable. *JLP*, 15(3):231–241, 1993.

[Tao *et al.*, 2010] Jiao Tao, Evren Sirin, Jie Bao, and Deborah L. McGuinness. Integrity constraints in OWL. In *AAAI'10*, 2010.