Mixed-World Reasoning with Existential Rules under Active-Domain Semantics

Meghyn Bienvenu, Pierre Bourhis

Mixed-World Reasoning with Existential Rules under Active-Domain Semantics

Meghyn Bienvenu[1] , **Pierre Bourhis**[2]

[1]CNRS, LaBRI, University of Bordeaux and Bordeaux INP, Talence, France
[2]CNRS, CRIStAL, University of Lille, Centrale Lille and INRIA Lille, Lille, France

Abstract

In this paper, we study reasoning with existential rules in a setting where some of the predicates may be closed (i.e., their content is fully specified by the data instance) and the remaining open predicates are interpreted under active-domain semantics. We show, unsurprisingly, that the main reasoning tasks (satisfiability and certainty / possibility of Boolean queries) are all intractable in data complexity in the general case. However, several positive (PTIME data) results are obtained for the linear fragment, and interestingly, these tractability results hold also for various extensions, e.g., with negated closed atoms and disjunctive rule heads. This motivates us to take a closer look at the linear fragment, exploring its expressivity and defining a fixpoint extension to approximate non-linear rules.

1 Introduction

There has been significant interest in recent years in ontology-mediated query answering (OMQA) [Bienvenu and Ortiz, 2015], in which data is enriched with an ontology that provides general domain knowledge. Ontologies are typically expressed using decidable fragments of first-order logic. While description logics (DLs) [Baader *et al.*, 2003] remain the most widely-studied formalism, existential rules (aka Datalog +/-) [Baget *et al.*, 2009; Baget *et al.*, 2011; Krötzsch and Rudolph, 2011; Calì *et al.*, 2012], which we adopt in this paper, have been drawing increasing interest, as they allow relations of arbitrary arity (unlike DLs which are restricted to unary and binary predicates) and generalize both Datalog [Abiteboul *et al.*, 1995] and tractable DLs like DL-Lite and $\mathcal{EL}$ [Calvanese *et al.*, 2007; Baader *et al.*, 2005].

The vast majority of work on OMQA adopts the *open world assumption*, whereby facts that are not present in the data instance are treated as unknown. Formally, each knowledge base (KB) $(\mathcal{R}, I)$, consisting of an ontology $\mathcal{R}$ and instance I, gives rise to a set of models, defined as the first-order structures that make true all facts in I and satisfy all rules (or axioms) in $\mathcal{R}$. When querying a KB, we are interested in obtaining the *certain answers*, i.e. the answers that hold for every model of the KB. By contrast, relational databases make the *closed-world assumption*, where each instance I is interpreted as the finite structure whose domain is equal to the *active domain* of I (i.e., the constants explicitly mentioned in I) and which makes true precisely the facts in I.

In practice, it seems natural to assume that applications may involve some predicates whose contents are fully known and others for which we have only partial information. This motivates the consideration of *mixed-world semantics*, where the set of predicates is partitioned into closed and open predicates, and models are required to coincide with the instance on the closed predicates. Mixed-world OMQA was first explored for DL ontologies [Lutz *et al.*, 2013; Lutz *et al.*, 2015; Ahmetaj *et al.*, 2016; Ngo *et al.*, 2016] and has only recently been studied for existential rules [Benedikt *et al.*, 2016]. Further variants of traditional OMQA semantics can be obtained by placing additional restrictions on the models. In particular, as the classical semantics considers arbitrary models with possibly infinite domains, it is interesting to consider the restriction to finite models [Ibáñez-García *et al.*, 2014; Gogacz *et al.*, 2018] , or models with fixed or bounded-size domains [Gaggl *et al.*, 2016; Rudolph and Schweizer, 2017].

The present paper combines elements of these different lines of research by exploring OMQA with existential rules under a hybrid mixed-world active-domain semantics, in which we can use both closed and open predicates, and the semantics is based upon active-domain models whose domains are equal to the active domain of the instance. Such a semantics is appropriate for scenarios in which all relevant constants are made available in the instance. A possible use case (formalized in Example 1 in simplified form), which served as a motivation for this work, is in analyzing the trajectories of people circulating in a geographically restricted area (e.g., industrial facility, closed medical facility) that is equipped with sensors and secure entry and exit (so that the set of people in the area is known at each timepoint).

The paper is structured as follows. In Section 3, we formally introduce our framework for mixed-world reasoning with existential rules under active-domain semantics, in particular, defining the certain and possible answers to conjunctive queries (CQs). In the following section, we analyze the data complexity of the three central reasoning tasks which are to determine whether a mixed-world knowledge base (MWKB) is satisfiable, and whether a Boolean CQ is certain or possible w.r.t. a MWKB. We show that all three tasks are intractable (NP- or coNP-complete in data complexity) for

MWKBs based upon arbitrary existential rules. However, for the linear fragment of our language (where rule bodies may contain at most one atom with an open predicate), we are able to establish several tractability results. Indeed, we can show that satisfiability, possibility of BCQs, and certainty are all PTIME-complete in data complexity, and moreover, the PTIME upper bounds remain valid in the presence of useful extensions, like (in)equality atoms, negated closed atoms, and disjunction in ruleheads. Motivated by these encouraging results, we investigate the linear fragment in more detail. In Section 4, we explore the expressive power of the linear fragment and prove our most technically challenging result, namely, that atomic queries coupled with mixed-world linear existential rules, extended with either closed negated atoms or disjunctive ruleheads and interpreted under either certain or possible active-domain semantics, can express all PTIME queries over ordered databases. Section 5 provides a general method of approximating unrestricted existential rules by means of linear rules augmented with a fixpoint semantics. We conclude the paper with a discussion of related and future work. Due to space restrictions, some proof details have been deferred to the appendix of the long version [Bienvenu and Bourhis, 2019].

2 Preliminaries

We assume the reader is familiar with standard notions from first-order logic. A *term* is either a constant or variable. An *atom* takes the form $P(t_1, \ldots, t_k)$, where P is a predicate (or relation) symbol of arity $k \geq 0$, and each t_i is a term. A *fact* is an atom that does not contain any variables, and an *instance* is a finite set of facts. The *active domain* of an instance I, denoted $\mathsf{adom}(I)$, is the set of constants occurring in I.

A *signature* Σ is a finite set of predicate symbols. A Σ-instance (resp. Σ-atom, Σ-fact) is an instance (resp. atom, fact) that only mentions predicates from Σ. The *projection of an instance I onto* Σ, denoted $I|_\Sigma$, is the set of Σ-facts in I. When S is a set of instances, we let $S|_\Sigma = \{I|_\Sigma \mid I \in S\}$.

A *conjunctive query (CQ)* is a conjunction of atoms in which some of the variables may be existentially quantified. *Atomic CQs* consists of a single atom, and *Boolean conjunctive queries (BCQs)* are CQs whose variables are all existentially quantified. We use $\mathsf{vars}(Q)$ to denote the set of variables in a CQ Q. We sometimes treat (B)CQs as sets of atoms, e.g., using $\alpha \in Q$ to say that α occurs as a conjunct of Q.

We recall that an *existential rule* (sometimes abbreviated to *rule*) is a first-order sentence of the form $\rho = \forall \vec{x} \varphi(\vec{x}) \rightarrow \exists \vec{y} \psi(\vec{x}, \vec{y})$, where $\varphi(\vec{x})$ and $\exists \vec{y} \psi(\vec{x}, \vec{y})$ are CQs (called respectively the *body* and *head* of ρ), and all variables in $\vec{x}$ occur in $\varphi(\vec{x})$. An *(existential) ruleset (over Σ)* is a finite set of existential rules (whose predicates are drawn from Σ). For brevity, we omit the initial universal quantifiers in rules, use commas in place of $\wedge$, and sometimes simplify to just $\varphi \rightarrow \psi$ (in which case the variables in $\mathsf{vars}(\psi) \setminus \mathsf{vars}(\varphi)$ are implicitly existentially quantified). As is common, we shall assume that ruleheads always consist of a single atom. This is w.l.o.g. since we can simulate conjunctive ruleheads by means of fresh predicates, obtaining a ruleset that is a conservative extension of the original one.

Existential rules are classically interpreted under first-order logic semantics. An existential rule $\varphi \rightarrow \psi$ is satisfied in a first-order structure $\mathfrak{J}$ with domain D if every variable assignment $h : \mathsf{vars}(\varphi) \rightarrow D$ that makes φ hold in $\mathfrak{J}$ can be extended to an assignment $h' : \mathsf{vars}(\varphi) \cup \mathsf{vars}(\psi) \rightarrow D$ such that ψ holds in $\mathfrak{J}$. We can extend the notion of satisfaction to instances as follows. With every instance I, we associate a finite first-order structure $\mathfrak{J}_I$ whose domain is $\mathsf{adom}(I)$ and such that every predicate P is interpreted as the set of tuples $\vec{c}$ such that $P(\vec{c}) \in I$. We then say that *rule ρ is satisfied in an instance I* (or that *I satisfies ρ*) if ρ is satisfied in $\mathfrak{J}_I$. Likewise, a *BCQ Q holds in instance I* if Q holds in $\mathfrak{J}_I$.

Datalog rules are existential rules with no existentially quantified variables, and a *Datalog program* is a finite set of Datalog rules. It is typical to partition the signature into *extensional predicates* that occur only in rule bodies, and *intensional predicates* that may occur both in bodies and heads. Given a Datalog program Π and an instance I over its extensional predicates, we denote by $\Pi(I)$ the minimal instance that contains I and satisfies all rules in Π. A *Datalog query* takes the form $(\Pi, Goal)$, where Π a Datalog program and $Goal$ an intensional predicate. When evaluated on instance I, such a query returns those tuples $\vec{a}$ such that $Goal(\vec{a}) \in \Pi(I)$. The preceding notions naturally extend to *semi-positive Datalog*, which allows negated extensional atoms in rule bodies.

3 Existential Rules with Closed Predicates and Active-Domain Semantics

This section introduces a framework for reasoning with existential rules in a setting where some of the predicates are declared as closed, and for the remaining open predicates, an active domain semantics is adopted, whereby only facts over the explicitly named constants are considered.

Formally, we will consider *mixed-world signatures* $\Sigma = (\Sigma_c, \Sigma_o)$, whose predicates are partitioned into a set Σ_c of *closed predicates*, and a set Σ_o of *open predicates*. Often we will only define Σ_c, leaving it implicit that all other predicates in Σ belong to Σ_o. A *mixed-world knowledge base (MWKB)* is a triple $(\Sigma, \mathcal{R}, I)$, where Σ is a mixed-world signature, $\mathcal{R}$ is an existential ruleset over Σ, and I is a Σ-instance. Note that $\mathcal{R}$ and I may use both open and closed predicates. Later in the paper, we will be especially interested in *linear existential rules*, which we define in our setting as existential rules whose *bodies contain at most one Σ_o-atom* (but can contain any number of Σ_c-atoms[1]).

We now define the *active-domain (AD) semantics* of MWKBs. A Σ-instance M is an *active-domain model* of a MWKB $(\Sigma, \mathcal{R}, I)$ if $\mathsf{adom}(M) = \mathsf{adom}(I)$, $M|_{\Sigma_c} = I|_{\Sigma_c}$, $I|_{\Sigma_o} \subseteq M$, and M satisfies every rule in $\mathcal{R}$. The set of active-domain models (henceforth abbreviated to models) of $(\Sigma, \mathcal{R}, I)$ is denoted $Mods_{AD}(\Sigma, \mathcal{R}, I)$. A MWKB $(\Sigma, \mathcal{R}, I)$ is *satisfiable under AD semantics* if $Mods_{AD}(\Sigma, \mathcal{R}, I) \neq \emptyset$. There are two natural ways of interpreting queries in our setting:

[1] Linear existential rules [Calì *et al.*, 2012] are usually restricted to a single body atom. Our definition is more general and in the spirit of linear Datalog rules [Abiteboul *et al.*, 1995].

- a BCQ Q is *certain w.r.t.* $(\Sigma, \mathcal{R}, I)$ *(under AD semantics)* if Q holds in every $M \in Mods_{AD}(\Sigma, \mathcal{R}, I)$;
- a BCQ Q is *possible w.r.t.* $(\Sigma, \mathcal{R}, I)$ *(under AD semantics)* if Q holds in some $M \in Mods_{AD}(\Sigma, \mathcal{R}, I)$.

We extend these notions to arbitrary CQs in the obvious way: a tuple $\vec{a}$ of constants from $\mathsf{adom}(I)$ is a *certain (resp. possible) answer* to a CQ $Q(\vec{x})$ with free variables $\vec{x}$ iff the BCQ $Q(\vec{a})$ obtained by replacing $\vec{x}$ by $\vec{a}$ is certain (resp. possible). Observe that a BCQ Q is not possible just in the case that its negation $\neg Q$ holds in every model, so the reasoning task of determining whether a BCQ is possible can be rephrased as determining whether a BCQ is certainly false. Note that because open predicates can be constrained by the closed predicates, not all open facts are possible (hence, some queries involving open predicates may be certainly false).

We will also consider a more general form of existential rule that allows for the following useful features: *(in)equality atoms* ($t = t'$, $t \neq t'$) in rule bodies and heads, *negated* Σ_c*-atoms* in rule bodies, and rule heads that take the form of a (possibly empty) *disjunction of existentially quantified atoms*. We will call these *extended (existential) rules*. An *extended linear rule* is an extended rule with at most one Σ_o-atom in its body, where (in)equality atoms are viewed as Σ_c-atoms. The notion of an extended rule holding in an instance is defined in the obvious way, following first-order logic semantics.

We conclude this section by an example that illustrates the use of our logic to reconstruct trajectories of users whose positions are recorded by different sensors in an area.

Example 1. The goal is to reconstruct paths of people circulating in a circumscribed area based upon sensor events. Every such event has an identifier e and provides the position p of a (possibly unidentified) user u at a certain time t. Nominative events are stored as facts $N(e, p, t, u)$, and all events (both nominative and anonymous) are stored as facts $E(e, p, t)$ in a generic event relation E. For the purposes of the example, we assume events occur at regular intervals (every Δ_t time units), and every user participates in some event (e.g., is detected by some sensor) at each timepoint (if this event identifies the user, then it is nominative, else anonymous). We further assume that $Next$ contains all pairs of events (e, e') that can feasibly occur in sequence i.e., the timepoint of e' directly follows that of e, and it is physically possible to move between their respective positions in Δ_t. Finally, we assume special nominative events containing the starting and ending positions of a user in the considered area, which can be found in relations S and F. Based upon the preceding information (stored in closed predicates), we construct possible trajectories by means of an open predicate Tr, where $Tr(u, e, e')$ means that user u transitions from event e to event e'; a second open predicate $Loc(u, p, t)$ states that u is at position p at time t. Our first two rules guess a subsequent event for every event up to the final event (note that the variable e'' with the guessed event is existentially quantified):

$$S(u, e, p, t), F(u, e', p', t'), t < t' \rightarrow Tr(u, e, e'')$$

$$Tr(u, e_0, e), E(e, p, t), F(u, e', p', t'), t < t' \rightarrow Tr(u, e, e'')$$

We use $Next$ to constrain the pairs of events in Tr,

$$Tr(u, e, e') \rightarrow Next(e, e')$$

and enforce that if the true position of a user is known at a given timepoint t (via a nominative event), then any selected event for that user at time t gives the same position:

$$Tr(u, e_0, e), E(e, p, t), N(e', p', t, u) \rightarrow p = p'$$

Loc-facts record the times and position of events in the guessed trajectories given by relation Tr:

$$Tr(u, e, e'), E(e', p, t) \rightarrow Loc(u, p, t)$$

and the final two rules forbid a user from being in more than one position at a time, and for distinct users being associated to the same position at the same time:

$$Loc(u, p, t), Loc(u, p', t) \rightarrow p = p'$$

$$Loc(u, p, t), Loc(u', p, t) \rightarrow u = u'$$

The first five rules are linear, while the last two are not. The certain (resp. possible) Loc-facts provide us with the sure (resp. potential) location of users at different timepoints, which could be used for visualizing and analyzing user trajectories. For example, displaying (or aggregating) the certain and possible locations of users (given by Loc) could help with understanding which sectors are the most frequented.

4 Complexity Analysis

We study the complexity of the three main reasoning tasks in our setting: satisfiability, certainty, and possibility. We focus on *data complexity*, which is measured in terms of the size of the instance. Our results are formulated for BCQs, but they apply equally well to the decision problem of testing whether a given tuple is a certain / possible answer.

Our first result shows that reasoning in the full language is intractable in data complexity, which is not surprising in light of prior negative results for mixed-world reasoning.

Theorem 1. *For (possibly extended) MWKBs with AD semantics, deciding if a BCQ is certain is coNP-complete in data complexity, while KB satisfiability and BCQ possibility are NP-complete in data complexity. The lower bounds hold for binary signatures and when the BCQ is a fact.*

Proof sketch. We focus on certainty, but the proofs are easily adapted for satisfiability and possibility. To show q is not certain w.r.t. $(\Sigma, \mathcal{R}, I)$, it suffices to guess an instance J with $\mathsf{adom}(J) = \mathsf{adom}(I)$, and verify that $I|_{\Sigma_o} \subseteq J$, J coincides with I on Σ_c, all rules in $\mathcal{R}$ are satisfied in J, and q does not hold in J. As J is of polynomial size in $|I|$, and we can verify rule satisfaction in PTIME w.r.t. $|J|$, we get an NP upper bound (data complexity) for non-certainty.

For the lower bound, we adapt a reduction from 2+2UNSAT sketched in [Lutz *et al.*, 2013]. Let $\mathcal{R}_{2+2}$ consist of:

- for $i \in \{1, 2\}$: $P_i(x, y) \rightarrow \exists z R(y, z)$, $P_i(x, y), R(y, z), F(z) \rightarrow U_i(x)$
- for $i \in \{3, 4\}$: $N_i(x, y) \rightarrow \exists z R(y, z)$, $N_i(x, y), R(y, z), T(z) \rightarrow U_i(x)$
- $R(y, x) \rightarrow B(x)$
- $\Big(S(w, x), \bigwedge_{1 \leq i \leq 4} U_i(x)\Big) \rightarrow Q(w)$

Given a CNF $\varphi = \lambda_1, \ldots, \lambda_n$ with $\lambda_i = v_1^i \vee v_2^i \vee \neg v_3^i \vee \neg v_4^i$, we construct the following instance I_{2+2}:

$$\{P(c_i, v_1^i), P(c_i, v_2^i), N(c_i, v_3^i), N(c_i, v_4^i) \mid 1 \leq i \leq n\}$$
$$\cup \{B(t), T(t), B(f), F(f)\} \cup \{S(a, c_i) \mid 1 \leq i \leq n\}$$

It is easily verified that, letting $\Sigma_c = \{B\}$, the fact $Q(a)$ is certain for $(\Sigma, \mathcal{R}_{2+2}, I_{2+2})$ iff φ is unsatisfiable. □

We next turn to linear rulesets. A straightforward adaptation of Theorem 1 shows that linearity does not always yield tractability, even for normally well-behaved classes of BCQs:

Theorem 2. *For linear MWKBs with AD semantics, deciding if an acyclic[2] BCQ is certain is coNP-hard in data complexity.*

Proof sketch. We adapt the reduction from Theorem 1 by (i) using the acyclic BCQ obtained by 'unfolding' $Q(a)$ w.r.t. the rules with head predicates Q and U_i, and then (ii) removing all non-linear rules from the ruleset $\mathcal{R}_{2+2}$. □

Nevertheless, we can show several positive results for linear rules, exploiting the following *maximal model property*:

Lemma 1. *Consider a signature $\Sigma = (\Sigma_c, \Sigma_o)$, an extended linear ruleset $\mathcal{R}$ over Σ, and Σ-instances I, N with $\mathsf{adom}(N) \subseteq \mathsf{adom}(I)$. If $\{M \in Mods_{AD}(\Sigma, \mathcal{R}, I) \mid M \cap N = \emptyset\}$ is non-empty, then it has a maximal element w.r.t. set inclusion, called the* maximal model of $(\Sigma, \mathcal{R}, I)$ *that omits N. Moreover, computing this model (or determining its non-existence) can be done in polynomial time in $|I|$.*

Proof. We outline a procedure for testing whether there is a model of $(\Sigma, \mathcal{R}, I)$ that omits N, and constructing the maximal such model when it exists. If $N \cap I \neq \emptyset$, then we immediately fail. Otherwise, we set J equal to the (unique) maximal Σ-instance with $\mathsf{adom}(J) = \mathsf{adom}(I)$ and $J \cap N = \emptyset$ that coincides with I on Σ_c. If J satisfies $\mathcal{R}$, then we have found the desired maximal model. Otherwise, repeat the following while J does not satisfy $\mathcal{R}$:

1. Pick a rule $\gamma \to \psi$ that is not satisfied in J, together with a variable assignment $h : \mathsf{vars}(\gamma) \to \mathsf{adom}(J)$ such that $h(\gamma)$ is satisfied in J but there is no extension h' of h to $\mathsf{vars}(\gamma) \cup \mathsf{vars}(\psi)$ such that $h'(\psi)$ holds in J.
2. If γ contains an atom $P(\vec{x})$ with $P \in \Sigma_o$ and $P(h(\vec{x})) \notin I$, then remove $P(h(\vec{x}))$ from J.
3. Else, halt and return 'no such model'.

If at some point J satisfies $\mathcal{R}$, halt and return J.

Correctness of this procedure can be established by an inductive argument, which shows that the current set J is always a superset of the desired maximal model if it exists (intuitively because every fact removal is 'forced'). Note that the initial J can be constructed in polynomial time in $|I|$, and there at most polynomially many iterations of the above steps, so the procedure runs in PTIME w.r.t. data complexity. □

As an immediate consequence of Lemma 1, we have:

[2]We recall that a BCQ Q is acyclic if the undirected graph (V_Q, E_Q) is acyclic, where V_Q contains the terms of Q, and E_Q contains (t, t') iff there is an atom of Q containing t and t'.

Theorem 3. *Satisfiability of extended linear MWKBs with AD semantics is in PTIME for data complexity.*

We can also use Lemma 1 to identify classes of queries for which the certainty problem is tractable.

Theorem 4. *For extended linear MWKBs with AD semantics, deciding if a fact is certain is in PTIME for data complexity. The same holds for ground CQs and for disjunctions of facts.*

Proof sketch. To test whether fact α is *not* certain, it suffices to determine the existence of a model that omits $N = \{\alpha\}$, a PTIME-checkable condition by Lemma 1. For a ground CQ $\alpha_1 \wedge \ldots \wedge \alpha_n$, we just perform this test for each α_i separately. Finally, note that a disjunction of facts $\alpha_1 \vee \ldots \vee \alpha_n$ is certain iff there is no model that omits $\{\alpha_1, \ldots, \alpha_n\}$. □

We now provide matching PTIME lower bounds for satisfiability and fact certainty. Recall that the latter problem is in $AC_0 \subsetneq$ PTIME for linear existential rules under classical semantics without closed predicates [Calì *et al.*, 2012].

Theorem 5. *For linear MWKBs with AD semantics, satisfiability and fact certainty are PTIME-hard in data complexity.*

Proof sketch. The proof is by reduction from the PTIME-complete Path Systems Accessibility (PSA) problem [Garey and Johnson, 1979]. A path system is a tuple (U, E, S, t) where U is a set of nodes, $E \subseteq U \times U \times U$ is an accessibility relation, $S \subseteq U$ is the set of source nodes, and $t \in U$ is the distinguished terminal node. The problem is then to determine whether $t \in \mathsf{access}(S)$, where $\mathsf{access}(S)$ is the least subset of U such that (i) $S \subseteq \mathsf{access}(S)$, and (ii) if $u_1, u_2 \in \mathsf{access}(S)$ and $(u_1, u_2, u_3) \in E$, then $u_3 \in \mathsf{access}(S)$.

Observe that $u \in \mathsf{access}(S)$ iff there is a labelled binary tree such that (i) the root is labelled u, (ii) every leaf is labelled by some $u' \in S$, and (iii) for every non-leaf node with label u_3, there is $(u_1, u_2, u_3) \in E$ such that the node's children are labelled u_1 and u_2. We remark that if such a witness tree exists, then there is one with depth is at most $|E|$. The idea for the reduction is to try to build such a witness tree.

Given a path system $PS = (U, E, S, t)$ with $|E| = m$, we construct (by a logspace transducer) the instance I_{PS} with:

- $Init(t, 0)$, with t the unique terminal node
- $R(u, u', u'', k, k+1)$, for $(u, u', u'') \in E$, $0 \leq k < m$
- $R(u, u, u, k, k+1)$, for each $u \in S$ and $0 \leq k < m$
- $R(u, u, u, m, m)$, for each $u \in S$

We set $\Sigma_c = \{Init, R\}$, and let $\mathcal{R}_{PS}$ consist of the rules:

- $Init(x, z) \to \exists y_1, y_2, z'\, T(y_1, y_2, x, z, z')$
- $T(y_1, y_2, x, z, z') \to R(y_1, y_2, x, z, z')$
- $T(y_1, y_2, x, z, z') \to \exists v_1, v_2, z'' T(v_1, v_2, y_1, z', z'')$
- $T(y_1, y_2, x, z, z') \to \exists v_1, v_2, z'' T(v_1, v_2, y_2, z', z'')$

The rough idea is that we guess a witness tree topdown, from the root t to the leaves, using the T-facts to record which triples from E were used. Correctness follows from the fact that every model of $(\Sigma, \mathcal{R}_{PS}, I_{PS})$ gives rise to a witness tree for $t \in \mathsf{access}(S)$ (constructed from the T-facts), and conversely, every such tree can be used to build a model. The

numbers occurring in the last two positions of T play a crucial role, by ordering the T-facts (allowing them to be assembled into a witness tree) and forcing each 'branch' to eventually stabilize in S (as once value m has been reached, we can only use T-facts of the form $T(u, u, u, m, m)$ with $u \in S$). It follows that $(\Sigma, \mathcal{R}_{PS}, I_{PS})$ is satisfiable iff $t \in \mathsf{access}(S)$. For certainty, we pick a fact $\alpha = Init(u, v) \notin I_{PS}$ and note that α is certain iff $(\Sigma, \mathcal{R}_{PS}, I_{PS})$ is unsatisfiable. □

For possibility, we obtain tractability for arbitrary BCQs:

Theorem 6. *For (extended) linear MWKBs with AD semantics, BCQ possibility is PTIME-complete for data complexity. The lower bound holds when the BCQ is a fact.*

Proof sketch. To decide if q is possible, it suffices to check whether q holds in the unique maximal model, which can be done in PTIME w.r.t. data complexity by Theorem 1.

The lower bound is by reduction from the PSA problem introduced earlier. The ruleset will consist of the following four linear rules, with $\Sigma_c = \{NR_0, B, In, Out\}$:

$$NR(x) \to NR_0(x) \quad NR(x), Out(y, x) \to \exists z NRIn(y, z)$$
$$NRIn(y, z) \to B(z) \quad In(y, z, x), NRIn(y, z) \to NR(x)$$

The input (U, E, S, t) is encoded by the following instance:

- $B(0), B(1)$, and $NR_0(u)$ for every $u \in U \setminus S$
- for every $e_i = (u_{i,0}, u_{i,1}, u_{i,2}) \in E$, the facts $In(e_i, 0, u_{i,0})$, $In(e_i, 1, u_{i,1})$, and $Out(e_i, u_{i,2})$

It can be verified that $u \in \mathsf{access}(S)$ iff $NR(u)$ is *not* possible for every $u \in U$, and in particular, when $u = t$, yielding a reduction from PSA to non-possibility. □

5 Expressive Power of Linear Fragment

We now study mixed-world linear existential rules under AD semantics from the perspective of query expressivity.

We recall that from an abstract perspective, a *k-ary query Q over a signature Σ* is a function that maps every Σ-instance I to a finite k-ary relation $Q(I)$ (containing the query answers). A query is called *generic* if it is invariant under renaming of constants[3], which implies in particular that $\mathsf{adom}(Q(I)) \subseteq \mathsf{adom}(I)$. With every k-ary query Q over Σ, we can associate the following recognition problem: given a Σ-instance I and k-tuple $\vec{a}$ of constants from $\mathsf{adom}(I)$, decide whether $\vec{a} \in Q(I)$. The class QPTIME consists of all generic queries whose recognition problem is computable in polynomial time w.r.t. data complexity.

Finding a logical language that precisely captures QPTIME is a major open problem in descriptive complexity. The following well-known result shows that the class of semi-positive Datalog queries captures QPTIME over so-called *ordered instances with min and max*, i.e., instances I with a binary predicate $Succ$ providing a successor relation among all constants in $\mathsf{adom}(I)$, and unary predicates Min and Max containing the smallest and largest constants.

Theorem 7 ([Papadimitriou, 1985])**.** *Over ordered instances with min and max, semi-positive Datalog captures* QPTIME.

[3]Queries that mention constants are not generic, but can be simulated using generic queries, see [Abiteboul *et al.*, 1995] for details.

Interestingly, we can show that our linear fragment, extended with either closed negated atoms or disjunctive ruleheads and interpreted with either certain or possible semantics, captures QPTIME over ordered instances. Formally, the next theorem concerns the classes of queries obtained by associating with every constant-free atomic query $A(\vec{x})$ and extended linear existential ruleset $\mathcal{R}$ over Σ the (generic) queries $Q_{A,\mathcal{R},c}$ and $Q_{A,\mathcal{R},p}$ that map I respectively to the certain and possible answers of $A(\vec{x})$ w.r.t. $(\Sigma, \mathcal{R}, I)$.

Theorem 8. *Over ordered instances, atomic queries coupled with mixed-world linear rulesets extended with either closed negated atoms or disjunctive ruleheads and interpreted under either the certain or possible AD semantics capture* QPTIME.

In the remainder of this section, we outline the key steps in the proof of Theorem 8. We start by noting that differently from Theorem 7, our result does not require the instance to provide the values min and max. This is because we can show how to compute these values from $Succ$ using our rules.

Suppose we are given a semi-positive Datalog program Π with extensional predicates $\Sigma_\Pi \supseteq \{Succ, Min, Max\}$ and k-ary output relation $Goal$. The core of the proof will be concerned with constructing a set of extended linear existential rules $\mathcal{R}_\Pi$ over Σ with $\Sigma_c = \Sigma_\Pi \setminus \{Min, Max\}$ such that for every ordered Σ_c-instance I and k-tuple of constants $\vec{a}$, the following statements are equivalent:

1. $Goal(\vec{a}) \in \Pi(I \cup \{Min(min), Max(max)\}$, with min and max the smallest and largest constants in I
2. $Ans(\vec{a})$ is certain w.r.t. $(\Sigma, \mathcal{R}_\Pi, I)$
3. $Ans(\vec{a})$ is possible w.r.t. $(\Sigma, \mathcal{R}_\Pi, I)$

To simplify the construction, we initially let $\mathcal{R}_\Pi$ consist of linear rules extended with both negated closed predicates and disjunctive ruleheads. Then in a final step, we show how we can get rid either of negated atoms or disjunctions.

Let us describe at a high level the ruleset $\mathcal{R}_\Pi$. We include rules that populate an open predicate $Tuple$ with all k-ary tuples of constants from the active domain and then guess for each tuple, whether it is an answer to the Datalog query:

$$Tuple(\vec{x}) \to Ans(\vec{x}) \vee NotAns(\vec{x})$$

To verify that these guesses are correct, we rely upon the fact that $\alpha \in \Pi(J)$ iff there is a *proof tree* for α [Abiteboul *et al.*, 1995], i.e., a tree whose root is labelled α, whose every leaf node is labelled with either an extensional fact β with $\beta \in J$ or a negated extensional fact $\neg\beta$ with $\beta \notin J$, and such that for each non-leaf node with label β and whose children have labels $\gamma_1, \ldots, \gamma_k$, the ground rule $\gamma_1, \ldots, \gamma_k \to \beta$ can be obtained from a rule in Π by instantiating its variables with constants from $\mathsf{adom}(J)$. It is known that the depth of a minimal proof tree is bounded by $p(|I|)$ for some polynomial function p. To verify $Ans(\vec{a})$, we follow a similar approach to the proof of Theorem 5 and guess a proof tree for $Goal(\vec{a})$ by working from root to leaves (failing if no such proof tree exists). To verify a fact $NotAns(\vec{a})$, we use a second set of rules to enforce that there is no proof tree for $Goal(\vec{a})$ of depth at most $p(n)$. Essentially, we guess topdown a 'no-proof' tree whose root is labelled $Ans(\vec{a})$, whose leaves are labelled by extensional facts *not* in J (here we used closed

negated atoms) or negations of extensional facts from J, and such that for every non-leaf node ν with label β and every ground instantiation $\gamma_1, \ldots, \gamma_k \rightarrow \beta$ of a rule in Π, there is a child of ν that is labelled with some γ_i. We implement counters that keep track of the depth of the guessed trees and fail if we exceed $p(n)$. By construction, $(\Sigma, \mathcal{R}_\Pi, I)$ is satisfiable. Moreover, while the models may differ on which trees are guessed, they always coincide on Ans and $NotAns$ facts. It follows that $Ans(\vec{a})$ is certain iff it is possible iff $Goal(\vec{a}) \in \Pi(I \cup \{Min(min), Max(max)\}$.

The rules implementing the preceding construction presume the existence of at least two values ($min \neq max$), which are used to implement binary counters. Thus, we must identify instances containing a single constant ($min = max$) and create a separate set of rules to compute the answers in this limit case. Briefly, the idea is to create a dedicated rule for each single-constant instance, using negated closed atoms to identify the contents of the input single-constant instance.

To complete the proof, we show how to simulate negated closed atoms using disjunction in ruleheads, and conversely, how to simulate disjunctive ruleheads using negated closed atoms. The latter translation only works for instances with at least two constants, but this is sufficient for our purposes, as disjunction only occurs in the rules that fire on instances with multiple constants. We point out that if we were to restrict to the class of instances with at least two constants, then both transformations are applicable and yield (plain) linear rules.

6 Fixpoint Extension of Linear Fragment

While the (extended) linear fragment has desirable computational properties, it cannot express some useful constructs, such as functionality (cf. final rule from Example 1). In this section we show how arbitrary rulesets can be approximated via a fixpoint-style extension of the linear fragment.

The idea is as follows. Given an arbitrary ruleset $\mathcal{R}$ over $\Sigma = (\Sigma_c, \Sigma_o)$, we can first compute the certain and possible facts using only the linear rules in $\mathcal{R}$, which yields a subset of the 'true' certain facts and a superset of the possible ones. We store these facts in new closed predicates R^{lb} and R^{ub}, use rules to link them to the original open predicates, create new linear rules from the non-linear rules by replacing all open atoms but one with closed R^{lb}-atoms, then recompute the certain and possible facts. Iterating this process until fixpoint allows us to obtain more refined approximations of the certain and possible facts of the original ruleset.

Formally, an (extended) linear^{fp} ruleset over $\Sigma = (\Sigma_c, \Sigma_o)$ is an (extended) linear ruleset over $\Sigma^+ = (\Sigma_c^+, \Sigma_o)$ where

$$\Sigma_c^+ = \Sigma_c \cup \{R^{ub}, R^{lb} \mid R \in \Sigma_o\}$$

and R^{lb}-atoms (resp. R^{ub}-atoms) may only occur positively and in rule bodies (resp. ruleheads). Given such a ruleset $\mathcal{R}$, we denote by $\mathcal{R}^+$ the ruleset

$$\mathcal{R} \cup \{R^{lb}(\vec{x}) \rightarrow R(\vec{x}), R(\vec{x}) \rightarrow R^{ub}(\vec{x}) \mid R \in \Sigma_o\}$$

(with $\vec{x}$ a tuple of distinct variables of the same arity as R). These rules formalize that R^{lb} (resp. R^{ub}) places a lower (resp. upper) bound on the set of R-facts.

Given a linear^{fp} ruleset $\mathcal{R}$ over $\Sigma = (\Sigma_c, \Sigma_o)$ and a Σ-instance I, we define inductively the set $Mods^k(\Sigma, \mathcal{R}, I)$ of models of $(\Sigma, \mathcal{R}, I)$ at each stage $k \geq 1$:

- $Mods^1(\Sigma, \mathcal{R}, I) = Mods_{AD}(\Sigma^+, \mathcal{R}^+, I_0)|_\Sigma$, where
$$I_0 = I \cup \{R^{ub}(\vec{a}) \mid R \in \Sigma_o \text{ of arity } n, \vec{a} \in \mathsf{adom}(I)^n\}$$
- $Mods^{k+1}(\Sigma, \mathcal{R}, I) = Mods_{AD}(\Sigma^+, \mathcal{R}^+, I_k)|_\Sigma$ where
$$I_k = I \cup \{R^{lb}(\vec{a}) \mid R(\vec{a}) \in Cert^k(\Sigma, \mathcal{R}, I)\} \cup \{R^{ub}(\vec{a}) \mid R(\vec{a}) \in Poss^k(\Sigma, \mathcal{R}, I)\}$$

where, for every $k \geq 1$, the set $Cert^k(\Sigma, \mathcal{R}, I)$ (resp. $Poss^k(\Sigma, \mathcal{R}, I)$) contains the Σ-facts over $\mathsf{adom}(I)$ that appear in every (resp. some) $M \in Mods^k(\Sigma, \mathcal{R}, I)$. The next lemma shows that these sets eventually converge to a fixpoint.

Lemma 2. *For every linear^{fp} ruleset $\mathcal{R}$ over Σ and Σ-instance I, $Mods^{j+1}(\Sigma, \mathcal{R}, I) \subseteq Mods^j(\Sigma, \mathcal{R}, I)$ for every $j \geq 1$, and there exists $k^* \geq 1$ such that $Mods^j(\Sigma, \mathcal{R}, I) = Mods^{k^*}(\Sigma, \mathcal{R}, I)$ for every $j \geq k^*$.*

Proof Sketch. We can show by simultaneous induction that (i) $Cert^{j-1}(\Sigma, \mathcal{R}, I) \subseteq Cert^j(\Sigma, \mathcal{R}, I)$, (ii) $Poss^j(\Sigma, \mathcal{R}, I) \subseteq Poss^{j-1}(\Sigma, \mathcal{R}, I)$, and (iii) $Mods^{j+1}(\Sigma, \mathcal{R}, I) \subseteq Mods^j(\Sigma, \mathcal{R}, I)$ for every $j \geq 1$. □

In what follows, we'll use $Mods^\infty(\Sigma, \mathcal{R}, I)$ to refer to the fixpoint $Mods^{k^*}(\Sigma, \mathcal{R}, I)$ from Lemma 2, and similarly for $Cert^\infty(\Sigma, \mathcal{R}, I)$ and $Poss^\infty(\Sigma, \mathcal{R}, I)$. We observe that the stage k^* where the fixpoint is reached is bounded polynomially in $|I|$. When combined with Theorems 4 and 6, this yields the following tractability result:

Theorem 9. *The problems of deciding whether a fact belongs to $Cert^\infty(\Sigma, \mathcal{R}, I)$ or to $Poss^\infty(\Sigma, \mathcal{R}, I)$ are both in PTIME for data complexity.*

Next consider an extended existential ruleset $\mathcal{R}$ over $\Sigma = (\Sigma_c, \Sigma_o)$. The *linearization* of $\mathcal{R}$, denoted $\mathcal{R}^{lin}$, is obtained by replacing each rule $\varepsilon \wedge \beta_1 \wedge \ldots \wedge \beta_m \rightarrow \gamma \in \mathcal{R}$, whose body Σ_o-atoms are $\beta_1, \ldots, \beta_m$, with the set of rules:

$$\varepsilon \wedge \bigwedge_{1 \leq j \leq m, j \neq i^*} \beta_j^{lb} \wedge \beta_{i^*} \rightarrow \gamma \qquad (1 \leq i^* \leq m)$$

where $\beta_i^{lb} = R^{lb}(\vec{t})$ when $\beta_i = R(\vec{t})$.

The following theorem shows that $\mathcal{R}^{lin}$, under the above fixpoint semantics, provides an under-approximation of the certain facts and an over-approximation of the possible facts.

Theorem 10. *Let $\mathcal{R}$ be an extended existential ruleset over Σ, and let $\mathcal{R}^{lin}$ be its linearization. For every Σ-instance I: $Mods_{AD}(\Sigma, \mathcal{R}, I) \subseteq Mods^\infty(\Sigma, \mathcal{R}^{lin}, I)$. In particular, this implies that for every fact α:*

- *$\alpha \in Cert^\infty(\Sigma, \mathcal{R}^{lin}, I) \Rightarrow \alpha$ is certain w.r.t. $(\Sigma, \mathcal{R}, I)$;*
- *α is possible w.r.t. $(\Sigma, \mathcal{R}, I) \Rightarrow \alpha \in Poss^\infty(\Sigma, \mathcal{R}^{lin}, I)$.*

The next result shows that linearization provides a non-trivial approximation in the sense that in general there is no instance-independent bound on the number of iterations needed to reach the fixpoint.

Lemma 3. *There exists a Σ-ruleset $\mathcal{R}$ such for every $j \geq 0$, there is a Σ-instance I such that $Mods^{j+1}(\Sigma, \mathcal{R}, I) \neq Mods^{j}(\Sigma, \mathcal{R}, I)$.*

Proof. Consider the following ruleset[4] $\mathcal{R}_{sat}$:

- $Clause(u) \rightarrow SatLit(u, x, y)$
- $SatLit(u, x, y) \rightarrow ClauseLit(u, x, y)$
- $SatLit(u, x, y) \rightarrow PickVal(x, y)$
- $PickVal(x, y) \rightarrow B(y)$
- $PickVal(x, y) \wedge PickVal(x, y') \rightarrow y = y'$

With every CNF $\varphi = c_1 \wedge \ldots \wedge c_n$ over $v_1, \ldots, v_m$, we associate the instance I_φ consisting of the following facts:

- $B(0), B(1)$
- $Clause(c_i)$ for $1 \leq i \leq n$
- $ClauseLit(c_i, v_j, 1)$ if c_i contains v_j
- $ClauseLit(c_i, v_j, 0)$ if c_i contains $\neg v_j$

We let Σ_{sat} contain all of the preceding predicates, where $Clause, ClauseLit, B$ are the only closed predicates. It can be verified that $(\Sigma_{sat}, \mathcal{R}_{sat}, I_\varphi)$ iff φ is satisfiable. Moreover, $PickVal(p_i, 1)$ (resp. $PickVal(p_i, 0)$) is certain iff every satisfying valuation of φ sets p_i to true (resp. false).

The linearization $(\mathcal{R}_{sat}^{lin})^+$ retains the first four rules of $\mathcal{R}_{sat}$ and adds new rules, the most relevant being:

- $PickVal^{lb}(x, y) \wedge PickVal(x, y') \rightarrow y = y'$
- $PickVal^{lb}(x, y) \rightarrow PickVal(x, y)$,
- $PickVal(x, y) \rightarrow PickVal^{ub}(x, y)$

Consider the sequence of propositional CNF ξ_i ($i \geq 1$), where $\xi_i = c_1 \wedge \ldots \wedge c_i$, $c_1 = p_1$, and $c_{\ell+1} = \neg p_\ell \vee p_{\ell+1}$ for $1 \leq \ell < i$. As every ξ_i is satisfiable, we know that $(\Sigma_{sat}, \mathcal{R}_{sat}, I_\varphi)$ is satisfiable. By Theorem 10, the same is true of $(\Sigma_{sat}, \mathcal{R}_{sat}^{lin}, I_{\xi_i})$. Moreover, $\xi_i \models p_\ell$ for every $1 \leq \ell \leq i$, so by the above, we have $PickVal(p_\ell, 1) \in Cert^{\infty}(\Sigma_{sat}, \mathcal{R}_{sat}^{lin}, I_{\xi_i})$ for all $1 \leq \ell \leq i$. A simple inductive argument shows that for every $i \geq 1$ and $1 \leq k \leq i$:

$$PickVal(p_\ell, 1) \in Cert^{k}(\Sigma_{sat}, \mathcal{R}_{sat}^{lin}, I_{\xi_i}) \quad \text{iff} \quad 1 \leq \ell \leq k$$

Hence, if $Cert^{k}(\Sigma_{sat}^{+}, \mathcal{R}_{sat}^{lin}, I_{\xi_i}) = Cert^{\infty}(\Sigma_{sat}^{+}, \mathcal{R}_{sat}^{lin}, I_{\xi_i})$, then $k \geq i$. Thus, by considering ξ_i for increasing values of i, we can delay the achievement of the fixpoint, meaning it cannot be bounded independently of the instance. □

A relevant question for future work is to analyze, either formally or experimentally, the quality of our approximation.

7 Concluding Remarks

In this paper, we investigated the problem of reasoning with existential rules under a hybrid mixed-world active-domain semantics. While querying in our setting is intractable in the general case, we identified an interesting and non-trivial fragment based upon linear rules that is PTIME-complete in data complexity, and in fact, captures all of QPTIME over ordered instances. To obtain a more widely applicable PTIME result, we provided a method of approximating non-linear rules by means of linear rules equipped with a fixpoint semantics.

We point out that a PTIME upper bound for possible BCQs with mixed-world semantics has been shown in [Benedikt *et al.*, 2016; Benedikt *et al.*, 2018] for a more restricted notion of linear rule (and without PTIME-hardness and QPTIME results). The key difference with [Benedikt *et al.*, 2018] and other works on mixed-world reasoning [Lutz *et al.*, 2013; Lutz *et al.*, 2015; Ahmetaj *et al.*, 2016; Ngo *et al.*, 2016] is that we adopt active-domain semantics, whereas these other works use classical semantics under which new values can appear in the open predicates. This has a very significant impact on the decidability and complexity of reasoning. For example, certainty of BCQs is decidable in our setting even for arbitrary existential rules, while it is undecidable under classical semantics; for frontier-guarded rules, certainty of BCQs was shown to be EXPTIME-complete in data complexity [Benedikt *et al.*, 2016], while it is coNP-complete in data complexity for our setting. A notable similarity is that our PTIME membership results employ a greatest fixpoint construction, which is in the same spirit as techniques used in [Benedikt *et al.*, 2018; Benedikt *et al.*, 2016]. Other non-trivial data tractability results have been obtained for fact certainty in the presence of linear disjunctive Datalog rules [Kaminski *et al.*, 2016], and for exact views given as LAV source-to-target rules [Benedikt *et al.*, 2018].

The present work can be continued in several directions. First, we can study the combined complexity of reasoning in our setting for different classes of existential rules, and the impact of using a fixed but succinctly represented domain, in the spirit of [Rudolph and Schweizer, 2017]. There are also many interesting questions related to expressive power and the (non)existence of translations to various Datalog extensions (e.g, disjunctive Datalog [Eiter *et al.*, 1997], or Datalog with non-deterministic choice [Abiteboul and Vianu, 1991]). We note that polynomial translations to disjunctive Datalog with negation have been proposed to compute certain facts w.r.t. mixed-world KBs in expressive description logics [Ahmetaj *et al.*, 2016], and a similar approach may be worth exploring in our setting. On the practical side, we have already begun applying our framework to integrating geolocalization data as illustrated by our example. Here the main challenge is to develop more efficient PTIME algorithms that do not require the construction of full maximal models, which although polynomial may be impractical for large domains. We would also like to allow for closed relations that are implicitly defined by PTIME-computable functions (e.g., $Next$ from Example 1 that could be defined using distances between locations) that are not computed and stored in advance but rather produced as needed.

[4] We have used an equality atom for convenience, but it is possible to modify the proof so that it works in the basic formalism.

Acknowledgements

We thank the reviewers for their detailed and constructive feedback. This work was partially supported by CNRS Momentum project "Managing Data without Leaks" and ANR project CQFD (ANR-18-CE23-0003).

References

[Abiteboul and Vianu, 1991] Serge Abiteboul and Victor Vianu. Non-determinism in logic-based languages. *Annals of Mathematics and Artificial Intelligence*, 3(2-4):151–186, 1991.

[Abiteboul *et al.*, 1995] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[Ahmetaj *et al.*, 2016] Shqiponja Ahmetaj, Magdalena Ortiz, and Mantas Simkus. Polynomial datalog rewritings for expressive description logics with closed predicates. In *Proceedings of IJCAI*, pages 878–885, 2016.

[Baader *et al.*, 2003] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.

[Baader *et al.*, 2005] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proceedings of IJCAI*, 2005.

[Baget *et al.*, 2009] Jean-François Baget, Michel Leclère, Marie-Laure Mugnier, and Eric Salvat. Extending decidable cases for rules with existential variables. In *Proceedings of IJCAI*, 2009.

[Baget *et al.*, 2011] Jean-François Baget, Michel Leclère, Marie-Laure Mugnier, and Eric Salvat. On rules with existential variables: Walking the decidability line. *Artificial Intelligence*, 175(9-10):1620–1654, 2011.

[Benedikt *et al.*, 2016] Michael Benedikt, Pierre Bourhis, Balder ten Cate, and Gabriele Puppis. Querying visible and invisible information. In *Proceedings of LICS*, 2016.

[Benedikt *et al.*, 2018] Michael Benedikt, Bernardo Cuenca Grau, and Egor V. Kostylev. Logical foundations of information disclosure in ontology-based data integration. *Artificial Intelligence*, 262:52–95, 2018.

[Bienvenu and Bourhis, 2019] Meghyn Bienvenu and Pierre Bourhis. Long version of this paper (with appendix). Available at http://www.labri.fr/perso/meghyn/papers/BieBou-IJCAI19.pdf, 2019.

[Bienvenu and Ortiz, 2015] Meghyn Bienvenu and Magdalena Ortiz. Ontology-mediated query answering with data-tractable description logics. In *Lecture Notes of the Reasoning Web Summer School*, volume 9203 of *LNCS*, pages 218–307. Springer, 2015.

[Calì *et al.*, 2012] Andrea Calì, Georg Gottlob, and Thomas Lukasiewicz. A general datalog-based framework for tractable query answering over ontologies. *Journal of Web Semantics*, 14:57–83, 2012.

[Calvanese *et al.*, 2007] 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. *Journal of Automated Reasoning*, 39(3):385–429, 2007.

[Eiter *et al.*, 1997] Thomas Eiter, Georg Gottlob, and Heikki Mannila. Disjunctive datalog. *ACM Transactions on Database Systems*, 22(3):364–418, 1997.

[Gaggl *et al.*, 2016] Sarah Alice Gaggl, Sebastian Rudolph, and Lukas Schweizer. Fixed-domain reasoning for description logics. In *Proceedings of ECAI*, 2016.

[Garey and Johnson, 1979] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[Gogacz *et al.*, 2018] Tomasz Gogacz, Yazmin Angélica Ibáñez-García, and Filip Murlak. Finite query answering in expressive description logics with transitive roles. In *Proceedings of KR*, 2018.

[Ibáñez-García *et al.*, 2014] Yazmin Angélica Ibáñez-García, Carsten Lutz, and Thomas Schneider. Finite model reasoning in horn description logics. In *Proceedings of KR*, 2014.

[Kaminski *et al.*, 2016] Mark Kaminski, Yavor Nenov, and Bernardo Cuenca Grau. Datalog rewritability of disjunctive datalog programs and non-horn ontologies. *Artificial Intelligence*, 236:90–118, 2016.

[Krötzsch and Rudolph, 2011] Markus Krötzsch and Sebastian Rudolph. Extending decidable existential rules by joining acyclicity and guardedness. In *Proceedings of IJCAI*, 2011.

[Lutz *et al.*, 2013] Carsten Lutz, Inanç Seylan, and Frank Wolter. Ontology-based data access with closed predicates is inherently intractable (sometimes). In *Proceedings of IJCAI*, 2013.

[Lutz *et al.*, 2015] Carsten Lutz, Inanç Seylan, and Frank Wolter. Ontology-mediated queries with closed predicates. In *Proceedings of IJCAI*, 2015.

[Ngo *et al.*, 2016] Nhung Ngo, Magdalena Ortiz, and Mantas Simkus. Closed predicates in description logics: Results on combined complexity. In *Proceedings of KR*, 2016.

[Papadimitriou, 1985] Christos H. Papadimitriou. A note the expressive power of prolog. *Bulletin of the EATCS*, 26, 1985.

[Rudolph and Schweizer, 2017] Sebastian Rudolph and Lukas Schweizer. Not too big, not too small... complexities of fixed-domain reasoning in first-order and description logics. In *Proceedings of EPIA*, 2017.