Introducing Nominals to the Combined Query Answering Approaches for $\mathcal{EL}$

Giorgio Stefanoni and **Boris Motik** and **Ian Horrocks**
Department of Computer Science, University of Oxford
Wolfson Building, Parks Road,
Oxford, OX1 3QD, UK

Abstract

So-called *combined approaches* answer a conjunctive query over a description logic ontology in three steps: first, they materialise certain consequences of the ontology and the data; second, they evaluate the query over the data; and third, they filter the result of the second phase to eliminate unsound answers. Such approaches were developed for various members of the DL-Lite and the $\mathcal{EL}$ families of languages, but none of them can handle ontologies containing nominals. In our work, we bridge this gap and present a combined query answering approach for $\mathcal{ELHO}^r_\bot$—a logic that contains all features of the OWL 2 EL standard apart from transitive roles and complex role inclusions. This extension is nontrivial because nominals require equality reasoning, which introduces complexity into the first and the third step. Our empirical evaluation suggests that our technique is suitable for practical application, and so it provides a practical basis for conjunctive query answering in a large fragment of OWL 2 EL.

Introduction

Description logics (DLs) (Baader et al. 2007) are a family of knowledge representation formalisms that underpin OWL 2 (Cuenca Grau et al. 2008)—an ontology language used in advanced information systems with many practical applications. Answering conjunctive queries (CQs) over ontology-enriched data sets is a core reasoning service in such systems, so the computational aspects of this problem have received a lot of interest lately. For expressive DLs, the problem is at least doubly exponential in query size (Glimm et al. 2008). The problem, however, becomes easier for the $\mathcal{EL}$ (Baader, Brandt, and Lutz 2005) and the DL-Lite (Calvanese et al. 2007) families of DLs, which provide the foundation for the OWL 2 EL and the OWL 2 QL profiles of OWL 2. An important goal of this research was to devise not only *worst-case optimal*, but also *practical* algorithms. The known approaches can be broadly classified as follows.

The first group consists of automata-based approaches for DLs such as OWL 2 EL (Krötzsch, Rudolph, and Hitzler 2007) and Horn-$\mathcal{SHOIQ}$ and Horn-$\mathcal{SROIQ}$ (Ortiz, Rudolph, and Simkus 2011). While worst-case optimal, these approaches are typically not suitable for practice since their best-case and worst-case performance often coincide.

The second group consists of rewriting-based approaches. Roughly speaking, these approaches rewrite the ontology and/or the query into another formalism, typically a union of conjunctive queries or a datalog program; the relevant answers can then be obtained by evaluating the rewriting over the data. Rewriting-based approaches were developed for members of the DL-Lite family (Calvanese et al. 2007; Artale et al. 2009), and the DLs $\mathcal{ELHIO}_\bot$ (Pérez-Urbina, Motik, and Horrocks 2010) and Horn-$\mathcal{SHIQ}$ (Eiter et al. 2012), to name just a few. A common problem, however, is that rewritings can be exponential in the ontology and/or query size. Although this is often not a problem in practice, such approaches are not worst-case optimal. An exception is the algorithm by Rosati (2007) that rewrites an $\mathcal{ELH}_\bot$ ontology into a datalog program of polynomial size; however, the algorithm also uses a nondeterministic step to transform the CQ into a tree-shaped one, and it is not clear how to implement this step in a goal-directed manner.

The third group consists of *combined approaches*, which use a three-step process: first, they augment the data with certain consequences of the ontology; second, they evaluate the CQ over the augmented data; and third, they filter the result of the second phase to eliminate unsound answers. The third step is necessary because, to ensure termination, the first step is unsound and may introduce facts that do not follow from the ontology; however, this is done in a way that makes the third step feasible. Such approaches have been developed for logics in the DL-Lite (Kontchakov et al. 2011) and the $\mathcal{EL}$ (Lutz, Toman, and Wolter 2009) families, and they are appealing because they are worst-case optimal and practical: only the second step is intractable (in query size), but it can be solved using well-known database techniques.

None of the combined approaches proposed thus far, however, handles *nominals*—concepts containing precisely one individual. Nominals are included in OWL 2 EL, and they are often used to state that all instances of a class have a certain property value, such as 'the sex of all men is male', or 'each German city is located in Germany'. In this paper we present a combined approach for $\mathcal{ELHO}^r_\bot$—the DL that covers all features of OWL 2 EL apart from transitive roles and complex role inclusions. To the best of our knowledge, this is the first combined approach that handles nominals. Our extension is nontrivial because nominals require equality reasoning, which increases the complexity of the first and the third step of the algorithm. In particular, nominals may introduce recursive dependencies in the filtering conditions used in the third phase; this is in contrast to the known combined approach for $\mathcal{EL}$ (Lutz, Toman, and Wolter 2009) in which filtering conditions are not recursive and can be incorporated into the input query. To solve this problem, our algorithm evaluates the original CQ and then uses a polynomial function to check the relevant conditions for each answer.

Following Krötzsch, Rudolph, and Hitzler (2008), instead of directly materialising the relevant consequences of the ontology and the data, we transform the ontology into a datalog program that captures the relevant consequences. Although seemingly just a stylistic issue, a datalog-based specification may be beneficial in practice: one can either materialise all consequences of the program bottom-up in advance, or one can use a top-down technique to compute only the consequences relevant for the query at hand. The latter can be particularly useful in informations systems that have read-only access to the data, or where data changes frequently.

We have implemented a prototypical system using our algorithm, and we carried out a preliminary empirical evaluation of (*i*) the blowup in the number of facts introduced by the datalog program, and (*ii*) the number of unsound answers obtained in the second phase. Our experiments show both of these numbers to be manageable in typical cases, suggesting that our algorithm provides a practical basis for answering CQs in an expressive fragment of OWL 2 EL.

The proofs of our technical results are provided in this paper's appendix.

Preliminaries

Logic Programming. We use the standard notions of variables, constants, function symbols, terms, atoms, formulas, and sentences (Fitting 1996). We often identify a conjunction with the set of its conjuncts. A substitution σ is a partial mapping of variables to terms; $\mathsf{dom}(\sigma)$ and $\mathsf{rng}(\sigma)$ are the domain and the range of σ, respectively; $\sigma|_S$ is the restriction of σ to a set of variables S; and, for α a term or a formula, $\sigma(\alpha)$ is the result of simultaneously replacing each free variable x occurring in α with $\sigma(x)$. A *Horn clause* C is an expression of the form $B_1 \wedge \ldots \wedge B_m \rightarrow H$, where H and each B_i are atoms. Such C is a *fact* if $m = 0$, and it is commonly written as H. Furthermore, C is *safe* if each variable occurring in H also occurs in some B_i. A *logic program* Σ is a finite set of safe Horn clauses; furthermore, Σ is a *datalog program* if each clause in Σ is function-free.

In this paper, we interpret a logic program Σ in a model that can be constructed bottom-up. The *Herbrand universe* of Σ is the set of all terms built from the constants and the function symbols occurring in Σ. Given an arbitrary set of facts B, let $\Sigma(B)$ be the smallest superset of B such that, for each clause $\varphi \rightarrow \psi \in \Sigma$ and each substitution σ mapping the variables occurring in the clause to the Herbrand universe of Σ, if $\sigma(\varphi) \subseteq B$, then $\sigma(\psi) \subseteq \Sigma(B)$. Let I_0 be the set of all facts occurring in Σ; for each $i \in \mathbb{N}$, let $I_{i+1} = \Sigma(I_i)$; and let $I = \bigcup_{i \in \mathbb{N}} I_i$. Then I is the *minimal Herbrand model* of Σ, and it is well known that I satisfies $\forall \vec{x}.C$ for each Horn clause $C \in \Sigma$ and $\vec{x}$ the vector of all variables occurring in C.

Type	Axiom		Clause
1	$\{a\} \sqsubseteq A$	$\rightsquigarrow$	$A(a)$
2	$A \sqsubseteq B$	$\rightsquigarrow$	$A(x) \rightarrow B(x)$
3	$A \sqsubseteq \{a\}$	$\rightsquigarrow$	$A(x) \rightarrow x \approx a$
4	$A_1 \sqcap A_2 \sqsubseteq A$	$\rightsquigarrow$	$A_1(x) \wedge A_2(x) \rightarrow A(x)$
5	$\exists R.A_1 \sqsubseteq A$	$\rightsquigarrow$	$R(x,y) \wedge A_1(y) \rightarrow A(x)$
6	$A_1 \sqsubseteq \exists R.A$	$\rightsquigarrow$	$A_1(x) \rightarrow R(x, f_{R,A}(x))$ $A_1(x) \rightarrow A(f_{R,A}(x))$
7	$R \sqsubseteq S$	$\rightsquigarrow$	$R(x,y) \rightarrow S(x,y)$
8	$\mathsf{range}(R, A)$	$\rightsquigarrow$	$R(x,y) \rightarrow A(y)$

Table 1: Transforming $\mathcal{ELHO}^r_\bot$ Axioms into Horn Clauses

In this paper we allow a logic program Σ to contain the equality predicate $\approx$. In first-order logic, $\approx$ is usually interpreted as the identity over the interpretation domain; however, $\approx$ can also be explicitly axiomatised (Fitting 1996). Let $\Sigma_\approx$ be the set containing clauses (1)–(3), an instance of clause (4) for each n-ary predicate R occurring in Σ and each $1 \leq i \leq n$, and an instance of clause (5) for each n-ary function symbol f occurring in Σ and each $1 \leq i \leq n$.

$$\rightarrow x \approx x \tag{1}$$

$$x_1 \approx x_2 \rightarrow x_2 \approx x_1 \tag{2}$$

$$x_1 \approx x_2 \wedge x_2 \approx x_3 \rightarrow x_1 \approx x_3 \tag{3}$$

$$R(\vec{x}) \wedge x_i \approx x_i' \rightarrow R(x_1, \ldots, x_i', \ldots, x_n) \tag{4}$$

$$x_i \approx x_i' \rightarrow f(\ldots, x_i, \ldots) \approx f(\ldots, x_i', \ldots) \tag{5}$$

The minimal Herbrand model of a logic program Σ that contains $\approx$ is the minimal Herbrand model of $\Sigma \cup \Sigma_\approx$.

Conjunctive Queries. A *conjunctive query* (CQ) is a formula $q = \exists \vec{y}.\psi(\vec{x}, \vec{y})$ with ψ a conjunction of function-free atoms over variables $\vec{x} \cup \vec{y}$. Variables $\vec{x}$ are the *answer variables* of q. Let $N_T(q)$ be the set of terms occurring in q.

Let τ be a substitution such that $\mathsf{rng}(\tau)$ contains only constants. Then, $\tau(q) = \exists \vec{z}.\tau(\psi)$, where $\vec{z}$ is obtained from $\vec{y}$ by removing each variable $y \in \vec{y}$ for which $\tau(y)$ is defined. Note that, according to this definition, non-free variables can also be replaced; for example, given $q = \exists y_1, y_2.R(y_1, y_2)$ and $\tau = \{y_2 \mapsto a\}$, we have $\tau(q) = \exists y_1.R(y_1, a)$.

Let Σ be a logic program, let I be the minimal Herbrand model of Σ, and let $q = \exists \vec{y}.\psi(\vec{x}, \vec{y})$ be a CQ that uses only the predicates occurring in Σ. A substitution π is a *candidate answer* for q in Σ if $\mathsf{dom}(\pi) = \vec{x}$ and $\mathsf{rng}(\pi)$ contains only constants; furthermore, such a π is a *certain answer* to q over Σ, written $\Sigma \models \pi(q)$, if a substitution τ exists such that $\mathsf{dom}(\tau) = \vec{x} \cup \vec{y}$, $\pi = \tau|_{\vec{x}}$, and $\tau(q) \subseteq I$.

Description Logic. DL $\mathcal{ELHO}^r_\bot$ is defined w.r.t. a signature consisting of mutually disjoint and countably infinite sets N_C, N_R, and N_I of *atomic concepts* (i.e., unary predicates), *roles* (i.e., binary predicates), and *individuals* (i.e., constants), respectively. Furthermore, for each individual $a \in N_I$, expression $\{a\}$ denotes a *nominal*—that is, a concept containing precisely the individual a. Also, we assume that $\top$ and $\bot$ are unary predicates (without any predefined meaning) not occurring in N_C. We consider only *normalised* knowledge bases, as it is well known (Baader, Brandt, and Lutz 2005) that each $\mathcal{ELHO}^r_\bot$ knowledge base can be normalised in polynomial time without affecting the answers to CQs. An $\mathcal{ELHO}^r_\bot$ *TBox* is a finite set of axioms of the form shown in the left-hand side of Table 1, where $A_{(i)} \in N_C \cup \{\top\}$, $B \in N_C \cup \{\top, \bot\}$, $R, S \in N_R$, and $a \in N_I$. An *ABox* $\mathcal{A}$ is a finite set of facts constructed using the symbols from $N_C \cup \{\top, \bot\}$, N_R, and N_I. Finally, an $\mathcal{ELHO}^r_\bot$ *knowledge base* (KB) is a tuple $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T}$ is an $\mathcal{ELHO}^r_\bot$ TBox $\mathcal{T}$ and an $\mathcal{A}$ is an ABox such that each predicate occurring in $\mathcal{A}$ also occurs in $\mathcal{T}$.

We interpret $\mathcal{K}$ as a logic program. Table 1 shows how to translate a TBox $\mathcal{T}$ into a logic program $\Xi(\mathcal{T})$. Moreover, let $\top(\mathcal{T})$ be the set of the following clauses instantiated for each atomic concept A and each role R occurring in $\mathcal{T}$.

$$A(x) \to \top(x) \quad R(x, y) \to \top(x) \quad R(x, y) \to \top(y)$$

A knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is translated into the logic program $\Xi(\mathcal{K}) = \Xi(\mathcal{T}) \cup \top(\mathcal{T}) \cup \mathcal{A}$. Then, $\mathcal{K}$ is *unsatisfiable* if $\Xi(\mathcal{K}) \models \exists y.\bot(y)$. Furthermore, given a conjunctive query q and a candidate answer π for q, we write $\mathcal{K} \models \pi(q)$ iff $\mathcal{K}$ is unsatisfiable or $\Xi(\mathcal{K}) \models \pi(q)$. Although somewhat nonstandard, our definitions of DLs are equivalent to the ones based on the standard denotational semantics (Baader et al. 2007). Given a candidate answer π for q, deciding whether $\Xi(\mathcal{K}) \models \pi(q)$ holds is NP-complete in *combined complexity*, and PTIME-complete in *data complexity* (Krötzsch, Rudolph, and Hitzler 2007).

Datalog Rewriting of $\mathcal{ELHO}^r_\bot$ TBoxes

For the rest of this section, we fix an arbitrary $\mathcal{ELHO}^r_\bot$ knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$. We next show how to transform $\mathcal{K}$ into a datalog program $\mathsf{D}(\mathcal{K})$ that can be used to check the satisfiability of $\mathcal{K}$. In the following section, we then show how to use $\mathsf{D}(\mathcal{K})$ to answer conjunctive queries.

Due to axioms of type 6 (cf. Table 1), $\Xi(\mathcal{K})$ may contain function symbols and is generally not a datalog program; thus, the evaluation of $\Xi(\mathcal{K})$ may not terminate. To ensure termination, we eliminate function symbols from $\Xi(\mathcal{K})$ using the technique by Krötzsch, Rudolph, and Hitzler (2008): for each $A \in N_C \cup \{\top\}$ and each $R \in N_R$ occurring in $\mathcal{T}$, we introduce a globally fresh and unique *auxiliary individual* $o_{R,A}$. Intuitively, $o_{R,A}$ represents all terms in the Herbrand universe of $\Xi(\mathcal{K})$ needed to satisfy the existential concept $\exists R.A$. Krötzsch, Rudolph, and Hitzler (2008) used this technique to facilitate taxonomic reasoning, while we use it to obtain a practical CQ answering algorithm. Please note that $o_{R,A}$ depends on both R and A, whereas in the known approaches such individuals depend only on A (Lutz, Toman, and Wolter 2009) or R (Kontchakov et al. 2011).

Definition 1. *Datalog program* $\mathsf{D}(\mathcal{T})$ *is obtained by translating each axiom of type other than 6 in the TBox* $\mathcal{T}$ *of* $\mathcal{K}$ *into a clause as shown in Table 1, and by translating each axiom* $A_1 \sqsubseteq \exists R.A$ *in* $\mathcal{T}$ *into clauses* $A_1(x) \to R(x, o_{R,A})$ *and* $A_1(x) \to A(o_{R,A})$*. Furthermore, the translation of* $\mathcal{K}$ *into datalog is given by* $\mathsf{D}(\mathcal{K}) = \mathsf{D}(\mathcal{T}) \cup \top(\mathcal{T}) \cup \mathcal{A}$.

Example 1. *Let* $\mathcal{T}$ *be the following* $\mathcal{ELHO}^r_\bot$ *TBox:*

$$\begin{array}{rr} KRC \sqsubseteq \exists taught.JProf & \exists taught.\top \sqsubseteq Course \\ Course \sqsubseteq \exists taught.Prof & \{kr\} \sqsubseteq KRC \\ Prof \sqsubseteq \exists advisor.Prof & KRC \sqsubseteq Course \\ JProf \sqsubseteq \{john\} & \mathsf{range}(taught, Prof) \end{array}$$

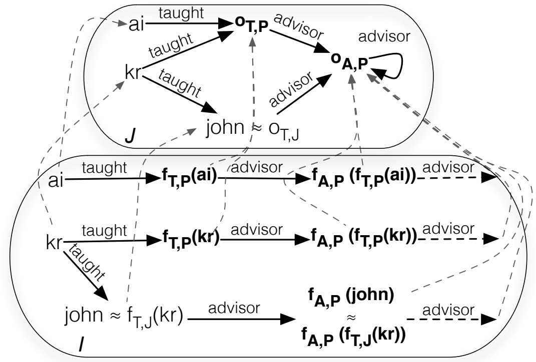

Figure 1: Representing the Models of $\Xi(\mathcal{K})$.

Then, $\mathsf{D}(\mathcal{T})$ *contains the following clauses:*

$$\begin{array}{ll} KRC(x) \to taught(x, o_{T,J}) & JProf(x) \to x \approx john \\ KRC(x) \to JProf(o_{T,J}) & taught(x, y) \to Course(x) \\ Course(x) \to taught(x, o_{T,P}) & KRC(kr) \\ Course(x) \to Prof(o_{T,P}) & KRC(x) \to Course(x) \\ Prof(x) \to advisor(x, o_{A,P}) & taught(x, y) \to Prof(y) \\ Prof(x) \to Prof(o_{A,P}) & \end{array}$$

◊

The following result straightforwardly follows from the definition of $\Xi(\mathcal{K})$ and $\mathsf{D}(\mathcal{K})$.

Proposition 2. *Program* $\mathsf{D}(\mathcal{K})$ *can be computed in time linear in the size of* $\mathcal{K}$.

Next, we prove that the datalog program $\mathsf{D}(\mathcal{K})$ can be used to decide the satisfiability of $\mathcal{K}$. To this end, we define a function δ that maps each term w in the Herbrand universe of $\Xi(\mathcal{K})$ to the Herbrand universe of $\mathsf{D}(\mathcal{K})$ as follows:

$$\delta(w) = \begin{cases} w & \text{if } w \in N_I, \\ o_{R,A} & \text{if } w \text{ is of the form } w = f_{R,A}(w'). \end{cases}$$

Let I and J be the minimal Herbrand models of $\Xi(\mathcal{K})$ and $\mathsf{D}(\mathcal{K})$, respectively. Mapping δ establishes a tight relationship between I and J as illustrated in the following example.

Example 2. *Let* $\mathcal{A} = \{Course(ai)\}$*, let* $\mathcal{T}$ *be as in Example 1, and let* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$*. Figure 1 shows a graphical representation of the minimal Herbrand models I and J of* $\Xi(\mathcal{K})$ *and* $\mathsf{D}(\mathcal{K})$*, respectively. The grey dotted lines show how* δ *relates the terms in I to the terms in J. For the sake of clarity, Figure 1 does not show the reflexivity of* $\approx$. ◊

Mapping δ is a homomorphism from I to J.

Lemma 3. *Let I and J be the minimal Herbrand models of* $\Xi(\mathcal{K})$ *and* $\mathsf{D}(\mathcal{K})$*, respectively. Mapping* δ *satisfies the following three properties for all terms* w' *and* w*, each* $B \in N_C \cup \{\top, \bot\}$*, and each* $R \in N_R$.

1. $B(w) \in I$ *implies* $B(\delta(w)) \in J$.
2. $R(w', w) \in I$ *implies* $R(\delta(w'), \delta(w)) \in J$.
3. $w' \approx w \in I$ *implies* $\delta(w') \approx \delta(w) \in J$.

For a similar result in the other direction, we need a couple of definitions. Let H be an arbitrary Herbrand model. Then,

$\mathsf{dom}(H)$ is the set containing each term w that occurs in H in at least one fact with a predicate in $N_C \cup \{\top, \bot\} \cup N_R$; note that, by this definition, we have $w \notin \mathsf{dom}(H)$ whenever w occurs in H only in assertions involving the $\approx$ predicate. Furthermore, aux_H is the set of all terms $w \in \mathsf{dom}(H)$ such that, for each term w' with $w \approx w' \in H$, we have $w' \notin N_I$. We say that the terms in aux_H are 'true' auxiliary terms—that is, they are not equal to an individual in N_I. In Figure 1, bold terms are 'true' auxiliary terms in I and J.

Lemma 4. *Let I and J be the minimal Herbrand models of $\Xi(\mathcal{K})$ and $\mathsf{D}(\mathcal{K})$, respectively. Mapping δ satisfies the following five properties for all terms w_1 and w_2 in $\mathsf{dom}(I)$, each $B \in N_C \cup \{\top, \bot\}$, and each $R \in N_R$.*

1. *$B(\delta(w_1)) \in J$ implies that $B(w_1) \in I$.*
2. *$R(\delta(w_1), \delta(w_2)) \in J$ and $\delta(w_2) \notin \mathsf{aux}_J$ imply that $R(w_1, w_2) \in I$.*
3. *$R(\delta(w_1), \delta(w_2)) \in J$ and $\delta(w_2) \in \mathsf{aux}_J$ imply that $\delta(w_2)$ is of the form $o_{P,A}$, that $R(w_1, f_{P,A}(w_1)) \in I$, and that a term w_1' exists such that $R(w_1', w_2) \in I$.*
4. *$\delta(w_1) \approx \delta(w_2) \in J$ and $\delta(w_2) \notin \mathsf{aux}_J$ imply that $w_1 \approx w_2 \in I$.*
5. *For each term u occurring in J, term $w \in \mathsf{dom}(I)$ exists such that $\delta(w) = u$.*

Lemmas 3 and 4 allow us to decide the satisfiability of $\mathcal{K}$ by answering a simple query over $\mathsf{D}(\mathcal{K})$, as shown in Proposition 5. The complexity claim is due to the fact that each clause in $\mathsf{D}(\mathcal{K})$ contains a bounded number of variables (Dantsin et al. 2001).

Proposition 5. *For $\mathcal{K}$ an arbitrary $\mathcal{ELHO}^r_\bot$ knowledge base, $\Xi(\mathcal{K}) \models \exists y.\bot(y)$ if and only if $\mathsf{D}(\mathcal{K}) \models \exists y.\bot(y)$. Furthermore, the satisfiability of $\mathcal{K}$ can be checked in time polynomial in the size of $\mathcal{K}$.*

Answering Conjunctive Queries

In this section, we fix a satisfiable $\mathcal{ELHO}^r_\bot$ knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ and a conjunctive query $q = \exists \vec{y}.\psi(\vec{x}, \vec{y})$. Furthermore, we fix I and J to be the minimal Herbrand models of $\Xi(\mathcal{K})$ and $\mathsf{D}(\mathcal{K})$, respectively.

While $\mathsf{D}(\mathcal{K})$ can be used to decide the satisfiability of $\mathcal{K}$, the following example shows that $\mathsf{D}(\mathcal{K})$ cannot be used directly to compute the answers to q.

Example 3. *Let $\mathcal{K}$ be as in Example 2, and let q_1, q_2, and q_3 be the following conjunctive queries:*

$$q_1 = taught(x_1, x_2)$$

$$q_2 = \exists y_1, y_2, y_3.\ taught(x_1, y_1) \wedge taught(x_2, y_2) \wedge advisor(y_1, y_3) \wedge advisor(y_2, y_3)$$

$$q_3 = \exists y.\ advisor(y, y)$$

Furthermore, let τ_i be the following substitutions:

$$\tau_1 = \{x_1 \mapsto kr,\ x_2 \mapsto o_{T,P}\}$$

$$\tau_2 = \{x_1 \mapsto kr,\ x_2 \mapsto ai,\ y_1 \mapsto o_{T,P},\ y_2 \mapsto o_{T,P},\ y_3 \mapsto o_{A,P}\}$$

$$\tau_3 = \{y \mapsto o_{A,P}\}$$

Finally, let each π_i be the projection of τ_i to the answer variables of q_i. Using Figure 1, one can readily check that $\mathsf{D}(\mathcal{K}) \models \tau_i(q_i)$, but $\Xi(\mathcal{K}) \not\models \pi_i(q_i)$, for each $1 \leq i \leq 3$. $\Diamond$

This can be explained by observing that J is a homomorphic image of I. Now homomorphisms preserve CQ answers (i.e., $\Xi(\mathcal{K}) \models \pi(q)$ implies $\mathsf{D}(\mathcal{K}) \models \pi(q)$), but they can also introduce unsound answers (i.e., $\mathsf{D}(\mathcal{K}) \models \pi(q)$ does not necessarily imply $\Xi(\mathcal{K}) \models \pi(q)$). This gives rise to the following notion of spurious answers.

Definition 6. *A substitution τ with $\mathsf{dom}(\tau) = \vec{x} \cup \vec{y}$ and $\mathsf{D}(\mathcal{K}) \models \tau(q)$ is a* spurious answer *to q if $\tau|_{\vec{x}}$ is not a certain answer to q over $\Xi(\mathcal{K})$.*

Based on these observations, we answer q over $\mathcal{K}$ in two steps: first, we evaluate q over $\mathsf{D}(\mathcal{K})$ and thus obtain an overestimation of the certain answers to q over $\Xi(\mathcal{K})$; second, for each substitution τ obtained in the first step, we eliminate spurious answers using a special function isSpur. We next formally introduce this function. We first present all relevant definitions, after which we discuss the intuitions. As we shall see, each query in Example 3 illustrates a distinct source of spuriousness that our function needs to deal with.

Definition 7. *Let τ be a substitution s.t. $\mathsf{dom}(\tau) = \vec{x} \cup \vec{y}$ and $\mathsf{D}(\mathcal{K}) \models \tau(q)$. Relation $\sim \subseteq N_T(q) \times N_T(q)$ for q, τ, and $\mathsf{D}(\mathcal{K})$ is the smallest reflexive, symmetric, and transitive relation closed under the* fork *rule, where $\mathsf{aux}_{\mathsf{D}(\mathcal{K})}$ is the set containing each individual u from $\mathsf{D}(\mathcal{K})$ for which no individual $c \in N_I$ exists such that $\mathsf{D}(\mathcal{K}) \models u \approx c$.*

$$(\mathsf{fork})\ \frac{s' \sim t'}{s \sim t}\quad \begin{array}{l} R(s, s') \text{ and } P(t, t') \text{ occur in } q \text{, and} \\ \tau(s') \in \mathsf{aux}_{\mathsf{D}(\mathcal{K})} \end{array}$$

Please note that the definition $\mathsf{aux}_{\mathsf{D}(\mathcal{K})}$ is actually a reformulation of the definition of aux_J, but based on the consequences of $\mathsf{D}(\mathcal{K})$ rather than the facts in J.

Relation $\sim$ is reflexive, symmetric, and transitive, so it is an equivalence relation, which allows us to normalise each term $t \in N_T(q)$ to a representative of its equivalence class using the mapping γ defined below. We then construct a graph G_{aux} that checks whether substitution τ matches 'true' auxiliary individuals in a way that cannot be converted to a match over 'true' auxiliary terms in I.

Definition 8. *Let τ and $\sim$ be as specified in Definition 7. Function $\gamma : N_T(q) \mapsto N_T(q)$ maps each term $t \in N_T(q)$ to an arbitrary, but fixed representative $\gamma(t)$ of the equivalence class of $\sim$ that contains t. Furthermore, the directed graph $G_{\mathsf{aux}} = \langle V_{\mathsf{aux}}, E_{\mathsf{aux}} \rangle$ is defined as follows.*

- *Set V_{aux} contains a vertex $\gamma(t) \in N_T(q)$ for each term $t \in N_T(q)$ such that $\tau(t) \in \mathsf{aux}_{\mathsf{D}(\mathcal{K})}$.*
- *Set E_{aux} contains an edge $\langle \gamma(s), \gamma(t) \rangle$ for each atom of the form $R(s, t)$ in q such that $\{\gamma(s), \gamma(t)\} \subseteq V_{\mathsf{aux}}$.*

Query q is aux-cyclic w.r.t. τ and $\mathsf{D}(\mathcal{K})$ *if G_{aux} contains a cycle; otherwise, q is* aux-acyclic w.r.t. τ and $\mathsf{D}(\mathcal{K})$.

We are now ready to define our function that checks whether a substitution τ is a spurious answer.

Definition 9. *Let τ and $\sim$ be as specified in Definition 7. Then, function $\mathsf{isSpur}(q, \mathsf{D}(\mathcal{K}), \tau)$ returns t if and only if at least one of the following conditions hold.*

(a) Variable $x \in \vec{x}$ exists such that $\tau(x) \notin N_I$.

(b) Terms s and t occurring in q exist such that $s \sim t$ and $\mathsf{D}(\mathcal{K}) \not\models \tau(s) \approx \tau(t)$.

(c) Query q is aux-cyclic w.r.t. τ and $\mathsf{D}(\mathcal{K})$.

We next discuss the intuition behind our definitions. We ground our discussion in minimal Herbrand models I and J, but our technique does not depend on such models: all conditions are stated as entailments that can be checked using an arbitrary sound and complete technique. Since $\mathcal{K}$ is an $\mathcal{ELHO}^r_\bot$ knowledge base, model I is *forest-shaped*: roughly speaking, the role assertions in I that involve at least one functional term are of the form $R(w_1, f_{R,A}(w_1))$ or $R(w_1, a)$ for $a \in N_I$; thus, I can be viewed as a family of directed trees whose roots are the individuals in N_I and whose edges point from parents to children or to the individuals in N_I. This is illustrated in Figure 1, whose lower part shows the the forest-model of the knowledge base from Example 3. Note that assertions of the form $R(w_1, a)$ are introduced via equality reasoning.

Now let τ be a substitution such that $\mathsf{D}(\mathcal{K}) \models \tau(q)$, and let $\pi = \tau|_{\vec{x}}$. If τ is not a spurious answer, it should be possible to convert τ into a substitution π^* such that $\pi = \pi^*|_{\vec{x}}$ and $\pi^*(q) \subseteq I$. Using the queries from Example 3, we next identify three reasons why this may not be possible.

First, τ may map an answer variable of q to an auxiliary individual, so by the definition π cannot be a certain answer to q; condition (a) of Definition 9 identifies such cases. Query q_1 and substitution τ_1 from Example 3 illustrate such a situation: $\tau_2(x_2) = o_{T,P}$ and $o_{T,P}$ is a 'true' auxiliary individual, so π_1 is not a certain answer to q_1.

The remaining two problems arise because model J is not forest-shaped, so τ might map q into J in a way that cannot be converted into a substitution π^* that maps q into I.

The second problem is best explained using substitution τ_2 and query q_2 from Example 3. Query q_2 contains a 'fork' $advisor(y_1, y_3) \wedge advisor(y_2, y_3)$. Now $\tau_2(y_3) = o_{A,P}$ is a 'true' auxiliary individual, and so it represents 'true' auxiliary terms $f_{A,P}(f_{T,P}(ai))$, $f_{A,P}(f_{T,P}(kr))$, and so on. Since I is forest-shaped, a match π_2^* for q in I obtained from τ_2 would need to map y_3 to one of these terms; let us assume that $\pi_2^*(y_3) = f_{A,P}(f_{T,P}(ai))$. Since I is forest-shaped and $f_{A,P}(f_{T,P}(ai))$ is a 'true' auxiliary term, this means that both y_1 and y_2 must be mapped to the same term (in both J and I). This is captured by the (fork) rule: in our example, the rule derives $y_1 \sim y_2$, and condition (b) of Definition 9 checks whether τ_2 maps y_1 and y_2 in a way that satisfies this constraint. Note that, due to role hierarchies, the rule needs to be applied to atoms $R(s, s')$ and $P(t, t')$ with $R \neq P$. Moreover, such constraints must be propagated further up the query. In our example, due to $y_1 \sim y_2$, atoms $taught(x_1, y_1) \wedge taught(x_2, y_2)$ in q_2 also constitute a 'fork', so the rule derives $x_1 \sim x_2$; now this allows condition (b) of Definition 9 to correctly identify τ_2 as spurious.

The third problem is best explained using substitution τ_3 and query q_3 from Example 3. Model J contains a 'loop' on individual $o_{A,P}$, which allows τ_3 to map q_3 into J. In contrast, model I is forest-shaped, and so the 'true' auxiliary terms that correspond to $o_{A,P}$ do not form loops. Condition (c) of Definition 9 detects such situations using the graph G_{aux}. The vertices of G_{aux} correspond to the terms of q that are matched to 'true' auxiliary individuals (mapping γ simply ensures that equal terms are represented as one vertex), and edges of G_{aux} correspond to the role atoms in q. Hence, if G_{aux} is cyclic, then the substitution π^* obtained from τ would need to match the query q over a cycle of 'true' auxiliary terms, which is impossible since I is forest-shaped.

Unlike the known combined approaches, our approach does not extend q with conditions that detect spurious answers. Due to nominals, the relevant equality constraints have a recursive nature, and they depend on both the substitution τ and on the previously derived constraints. Consequently, filtering in our approach is realised as postprocessing; furthermore, to ensure correctness of our filtering condition, auxiliary individuals must depend on both a role and an atomic concept. The following theorem proves the correctness of our approach.

	Individuals (% in $\mathsf{aux}_{\mathsf{D}(\mathcal{K})}$)	Unary facts (% over $\mathsf{aux}_{\mathsf{D}(\mathcal{K})}$)	Binary facts (% over $\mathsf{aux}_{\mathsf{D}(\mathcal{K})}$)
L-5	100848	169079	296941
Mat.	100868 (0.01)	309350 (0.01)	632489 (49.2)
L-10	202387	339746	598695
Mat.	202407 (0.01)	621158 (0.01)	1277575 (49.3)
L-20	426144	714692	1259936
Mat.	426164 (0.01)	1304815 (0.01)	2691766 (49.3)
SEM	17945	17945	47248
Mat.	17953 (0.04)	25608 (0.03)	76590 (38.3)

Table 2: Size of the materialisations.

Theorem 10. *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a satisfiable $\mathcal{ELHO}^r_\bot$ KB, let $q = \exists \vec{y}.\psi(\vec{x}, \vec{y})$ be a CQ, and let $\pi : \vec{x} \mapsto N_I$ be a candidate answer for q. Then, $\Xi(\mathcal{K}) \models \pi(q)$ iff a substitution τ exists such that $\mathsf{dom}(\tau) = \vec{x} \cup \vec{y}$, $\tau|_{\vec{x}} = \pi$, $\mathsf{D}(\mathcal{K}) \models \tau(q)$, and $\mathsf{isSpur}(q, \mathsf{D}(\mathcal{K}), \tau) = \mathsf{f}$.*

Furthermore, $\mathsf{isSpur}(q, \mathsf{D}(\mathcal{K}), \tau)$ can be evaluated in polynomial time, so the main source of complexity in our approach is in deciding whether $\mathsf{D}(\mathcal{K}) \models \tau(q)$ holds. This gives rise to the following result.

Theorem 11. *Deciding whether $\mathcal{K} \models \pi(q)$ holds can be implemented in nondeterministic polynomial time w.r.t. the size of $\mathcal{K}$ and q, and in polynomial time w.r.t. the size of $\mathcal{A}$.*

Evaluation

To gain insight into the practical applicability of our approach, we implemented our technique in a prototypical system. The system uses HermiT, a widely used ontology reasoner, as a datalog engine in order to materialise the consequences of $\mathsf{D}(\mathcal{K})$ and evaluate q. The system has been implemented in Java, and we ran our experiments on a MacBook Pro with 4GB of RAM and an Intel Core 2 Duo 2.4 Ghz processor. We used two ontologies in our evaluation, details of which are given below. The ontologies, queries, and the prototype system are all available online at http://www.cs.ox.ac.uk/isg/tools/KARMA/.

The LSTW benchmark (Lutz et al. 2012) consists of an OWL 2 QL version of the LUBM ontology (Guo, Pan, and Heflin 2005), queries $q^l_1, \ldots, q^l_{11}$, and a data generator. The LSTW ontology extends the standard LUBM ontology with several axioms of type 6 (see Table 1). To obtain an $\mathcal{ELHO}^r_\bot$ ontology, we removed inverse roles and datatypes, added 11 axioms using 9 freshly introduced nominals, and added one

LSTW	q_1^l Tot (%)	q_2^l Tot (%)	q_3^l Tot (%)	q_5^l Tot (%)	q_8^l Tot (%)	q_9^l Tot (%)	q_{10}^l Tot (%)
L-5	116K	3.7M	10	28K	13K	1K	12K
L-10	233K (4.0)	32M (100.0)	22 (0.0)	57K (0.0)	26K (26.0)	2K (0.0)	25K (74.5)
L-20	487K	170M	43	121K	55K	4K	53K

	q_1^s Tot (%)	q_2^s Tot (%)	q_3^s Tot (%)	q_4^s Tot (%)	q_5^s Tot (%)	q_6^s Tot (%)	q_7^s Tot (%)	q_8^s Tot (%)	q_9^s Tot (%)
SEMINTEC	7 (0.0)	53 (0.0)	16 (0.0)	12 (0.0)	31 (0.0)	838K (55.4)	5K (0.0)	5K (54.3)	13K (33.3)

Table 3: Total number of answers and ratio spurious to answers. In Table LSTW, the ratio is stable for each data set.

axiom of type 4 (see Table 1). These additional axioms resemble the ones in Example 1, and they were designed to test equality reasoning. The resulting signature consists of 132 concepts, 32 roles, and 9 nominals, and the ontology contains 180 axioms. From the 11 LSTW queries, we did not consider queries q_4^l, q_6^l, q_7^l, and q_{11}^l because their result sets were empty: q_4^l relies on existential quantification over inverse roles, and the other three are empty already w.r.t. the original LSTW ontology. Query q_2^l is similar to query q_2 from Example 3, and it was designed to produce only spurious answers and thus stress the system. We generated data sets with 5, 10 and 20 universities. For each data set, we denote with L-i the knowledge base consisting of our $\mathcal{ELHO}_\bot^r$ ontology and the ABox for i universities (see Table 2).

SEMINTEC is an ontology about financial services developed within the SEMINTEC project at the University of Poznan. To obtain an $\mathcal{ELHO}_\bot^r$ ontology, we removed inverse roles, role functionality axioms, and universal restrictions, added nine axioms of type 6 (see Table 1), and added six axioms using 4 freshly introduced nominals. The resulting ontology signature consists of 60 concepts, 16 roles, and 4 nominals, and the ontology contains 173 axioms. Queries q_1^s–q_5^s are tree-shaped queries used in the SEMINTEC project, and we developed queries q_6^s–q_9^s ourselves. Query q_6^s resembles query q_2^l from LSTW, and queries q_8^s and q_9^s were designed to retrieve a large number of answers containing auxiliary individuals, thus stressing condition (a) of Definition 9. Finally, the SEMINTEC ontology comes with a data set consisting of approximately 65,000 facts concerning 18,000 individuals (see row SEM in Table 2).

The practicality of our approach, we believe, is determined mainly by the following two factors. First, the number of facts involving auxiliary individuals introduced during the materialisation phase should not be 'too large'. Table 2 shows the materialisation results: the first column shows the number of individuals before and after materialisation and the percentage of 'true' auxiliary individuals, the second column shows the number of unary facts before and after materialisation and the percentage of facts involving a 'true' auxiliary individual, and the third column does the same for binary facts. As one can see, for each input data set, the materialisation step introduces few 'true' auxiliary individuals, and the number of facts at most doubles. The number of unary facts involving a 'true' auxiliary individual does not change with the size of the input data set, whereas the number of such binary facts increases by a constant factor. This is because, in clauses of type 6, atoms $A(o_{R,A})$ do not contain a variable, whereas atoms $R(x, o_{R,A})$ do.

Second, evaluating q over $\mathsf{D}(\mathcal{K})$ should not produce too many spurious answers. Table 3 shows the total number of answers for each query—that is, the number of answers obtained by evaluating the query over $\mathsf{D}(\mathcal{K})$; furthermore, the table also shows what percentage of these answers are spurious. Queries q_2^l, q_{10}^l, q_6^s, and q_8^s retrieve a significant percentage of spurious answers. However, only query q_2^l has proven to be challenging for our system due to the large number of retrieved answers, with an evaluation time of about 40 minutes over the largest knowledge base (L-20). Surprisingly, q_1^l also performed rather poorly despite a low number of spurious answers, with an evaluation time of about 20 minutes for L-20. All other queries were evaluated in at most a few seconds, thus suggesting that queries q_1^l and q_2^l are problematical mainly because HermiT does not implement query optimisation algorithms typically used in relational databases.

Conclusion

We presented the first combined technique for answering conjunctive queries over DL ontologies that include nominals. A preliminary evaluation suggests the following. First, the number of materialised facts over 'true' anonymous individuals increases by a constant factor with the size of the data. Second, query evaluation results have shown that, while some cases may be challenging, in most cases the percentage of answers that are spurious is manageable. Hence, our technique provides a practical CQ answering algorithm for a large fragment of OWL 2 EL.

We anticipate several directions for our future work. First, we would like to investigate the use of top-down query evaluation techniques, such as magic sets (Abiteboul, Hull, and Vianu 1995) or SLG resolution (Chen and Warren 1993). Second, tighter integration of the detection of spurious answers with the query evaluation algorithms should make it possible to eagerly detect spurious answers (i.e., before the query is fully evaluated). Lutz et al. (2012) already implemented a filtering condition as a user-defined function in a database, but it is unclear to what extent such an implementation can be used to optimise query evaluation. Finally, we would like to extend our approach to all of OWL 2 EL.

Acknowledgements

This work was supported by the Royal Society; Alcatel-Lucent; the EU FP7 project OPTIQUE; and the EPSRC projects ExODA, MASI[3], and QueRe.

References

Abiteboul, S.; Hull, R.; and Vianu, V. 1995. *Foundations of Databases*. Addison-Wesley.

Artale, A.; Calvanese, D.; Kontchakov, R.; and Zakharyaschev, M. 2009. The DL-Lite Family and Relations. *Journal of Artificial Intelligence Research* 36:1–69.

Baader, F.; Calvanese, D.; McGuinness, D.; Nardi, D.; and Patel-Schneider, P. F., eds. 2007. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press. ISBN 9780511717383.

Baader, F.; Brandt, S.; and Lutz, C. 2005. Pushing the $\mathcal{EL}$ Envelope. In Kaelbling, L. P., and Saffiotti, A., eds., *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005)*, 364–369. Edinburgh, UK: Morgan Kaufmann Publishers.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2007. Tractable Reasoning and Efficient Query Answering in Description Logics: The DL-Lite Family. *Journal of Automated Reasoning* 9(3):385–429.

Chen, W., and Warren, D. S. 1993. Query evaluation under the well-founded semantics. In *Proceedings of the 12th ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, PODS '93, 168–179. New York, NY, USA: ACM.

Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; and Stein, C. 2009. *Introduction to Algorithms (3. ed.)*. MIT Press.

Cuenca Grau, B.; Horrocks, I.; Motik, B.; Parsia, B.; Patel-Schneider, P.; and Sattler, U. 2008. OWL 2: The next step for OWL. *Journal of Web Semantics* 6(4):309–322.

Dantsin, E.; Eiter, T.; Gottlob, G.; and Voronkov, A. 2001. Complexity and expressive power of logic programming. *ACM Computing Surveys* 33(3):374–425.

Eiter, T.; Ortiz, M.; Simkus, M.; Tran, T.-K.; and Xiao, G. 2012. Query Rewriting for Horn-SHIQ Plus Rules. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence, (AAAI 2012)*. AAAI Press.

Fitting, M. 1996. *First-order logic and automated theorem proving (2nd ed.)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Glimm, B.; Horrocks, I.; Lutz, C.; and Sattler, U. 2008. Conjunctive Query Answering for the Description Logic $\mathcal{SHIQ}$. *Journal of Artificial Intelligence Research* 31:151–198.

Guo, Y.; Pan, Z.; and Heflin, J. 2005. LUBM: A benchmark for OWL knowledge base systems. *Journal of Web Semantics* 3(2–3):158–182.

Kontchakov, R.; Lutz, C.; Toman, D.; Wolter, F.; and Zakharyaschev, M. 2011. The Combined Approach to Ontology-Based Data Access. In Walsh, T., ed., *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*. AAAI Press.

Krötzsch, M.; Rudolph, S.; and Hitzler, P. 2007. Conjunctive queries for a tractable fragment of OWL 1.1. In Aberer, K.; Choi, K.-S.; Noy, N.; Allemang, D.; Lee, K.-I.; Nixon, L.; Golbeck, J.; Mika, P.; Maynard, D.; Mizoguchi, R.; Schreiber, G.; and Cudré-Mauroux, P., eds., *Proceedings of the 6th International Semantic Web Conference (ISWC'07)*, volume 4825 of *LNCS*, 310–323. Springer.

Krötzsch, M.; Rudolph, S.; and Hitzler, P. 2008. ELP: Tractable rules for OWL 2. In Sheth, A.; Staab, S.; Dean, M.; Paolucci, M.; Maynard, D.; Finin, T.; and Thirunarayan, K., eds., *Proceedings of the 7th International Semantic Web Conference (ISWC'08)*, volume 5318 of *LNCS*, 649–664. Springer.

Lutz, C.; Seylan, I.; Toman, D.; and Wolter, F. 2012. The Combined Approach to OBDA: Taming Role Hierarchies using Filters. In Fokoue, A.; Liebig, T.; Goodman, E.; Weaver, J.; Urbani, J.; and Mizell, D., eds., *Proceedings of the Joint Workshop on Scalable and High-Performance Semantic Web Systems (SSWS+HPCSW 2012)*, volume 943 of *CEUR Workshop Proceedings*, 16–31. CEUR-WS.org.

Lutz, C.; Toman, D.; and Wolter, F. 2009. Conjunctive Query Answering in the Description Logic $\mathcal{EL}$ Using a Relational Database System. In Boutilier, C., ed., *Proceedings of the 21st International Joint Conference on Artificial Intelligence, (IJCAI 2009)*, 2070–2075. AAAI Press.

Ortiz, M.; Rudolph, S.; and Simkus, M. 2011. Query Answering in the Horn Fragments of the Description Logics $\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$. In Walsh, T., ed., *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*, 1039–1044. Barcelona, Spain: AAAI Press.

Pérez-Urbina, H.; Motik, B.; and Horrocks, I. 2010. Tractable Query Answering and Rewriting under Description Logic Constraints. *Journal of Applied Logic* 8(2):186–209.

Rosati, R. 2007. On Conjunctive Query Answering in $\mathcal{EL}$. In Calvanese, D.; Franconi, E.; Haarslev, V.; Lembo, D.; Motik, B.; Turhan, A.-Y.; and Tessaris, S., eds., *Proceedings of the 20th International Workshop on Description Logics (DL-2007)*, CEUR Workshop Proceedings. CEUR-WS.org.

Additional Proofs

Proof of Lemma 3

Lemma 3. *Let I and J be the minimal Herbrand models of $\Xi(\mathcal{K})$ and $\mathsf{D}(\mathcal{K})$, respectively. Mapping δ satisfies the following three properties for all terms w' and w, each $B \in N_C \cup \{\top, \bot\}$, and each $R \in N_R$.*

1. *$B(w) \in I$ implies $B(\delta(w)) \in J$.*
2. *$R(w', w) \in I$ implies $R(\delta(w'), \delta(w)) \in J$.*
3. *$w' \approx w \in I$ implies $\delta(w') \approx \delta(w) \in J$.*

Proof. Let $I_0, I_1, \ldots$ be the sequence of sets used to construct I. We show by induction on n that each I_n satisfies the properties.

Base case. Consider I_0 and an arbitrary fact $H \in I_0$. Each term occurring in H is contained in N_I. Moreover, H is a fact from $\Xi(\mathcal{K})$ and, by definition, it is also a fact from $\mathsf{D}(\mathcal{K})$. Now δ is the identity over N_I, and J satisfies H, so properties 1 and 2 hold. Property 3 holds vacuously since I_0 does not contain facts with the equality predicate.

Inductive step. Consider an arbitrary $n \in \mathbb{N}$ and assume that I_n satisfies properties 1–3; we show that the same holds for I_{n+1}. Towards this goal, we consider the different clauses in $\Xi(\mathcal{K}) \cup \Xi(\mathcal{K})_\approx$ that can derive fresh facts from I_n. We distinguish the following two cases.

First, consider an arbitrary datalog clause of the form $\varphi \to \psi$ from $\Xi(\mathcal{K}) \cup \Xi(\mathcal{K})_\approx$. Let σ be an arbitrary substitution mapping variables occurring in the clause to the terms in the Herbrand universe of $\Xi(\mathcal{K})$ such that $\sigma(\varphi) \subseteq I_n$, so the clause derives $\sigma(\psi) \in I_{n+1}$. Let σ' be the substitution defined such that $\sigma'(x) = \delta(\sigma(x))$ for each variable x occurring in the clause. By the inductive hypothesis, we have $\sigma'(\varphi) \subseteq J$. Furthermore, by the definition of $\mathsf{D}(\mathcal{K})$, we have that $\mathsf{D}(\mathcal{K}) \cup \mathsf{D}(\mathcal{K})_\approx$ contains $\varphi \to \psi$. Finally, since J satisfies $\varphi \to \psi$, we have $\sigma'(\psi) \in J$, as required.

Second, consider arbitrary clauses from $\Xi(\mathcal{K})$ of the form $A_1(x) \to R(x, f_{R,A}(x))$ and $A_1(x) \to A(f_{R,A}(x))$, and assume that $A_1(w) \in I_n$; hence, these clauses derive $\{R(w, f_{R,A}(w)),\ A(f_{R,A}(w))\} \subseteq I_{n+1}$. By the inductive hypothesis, we have $A_1(\delta(w)) \in J$. Furthermore, by the definition of δ, we have that $\delta(f_{R,A}(w)) = o_{R,A}$. Moreover, by the definition of program $\mathsf{D}(\mathcal{K})$, the program contains clauses $A_1(x) \to R(x, o_{R,A})$ and $A_1(x) \to A(o_{R,A})$. Finally, model J satisfies both of these clauses, so we have $\{R(\delta(w), o_{R,A})\ A(o_{R,A})\} \subseteq J$, as required. □

Proof of Lemma 4

In order to prove Lemma 4, we use the properties from Lemmas 12 and 13.

Lemma 12. *For each term w_2, each role $R \in N_R$, and each concept $A \in N_C \cup \{\top\}$, if $f_{R,A}(w_2) \in \mathsf{dom}(I)$, then $\{R(w_2, f_{R,A}(w_2)),\ A(f_{R,A}(w_2))\} \subseteq I$.*

Proof. Let $I_0, I_1, \ldots$ be the sequence used to construct I; we assume w.l.o.g. that each I_{n+1} is obtained from I_n by applying just one clause type. We show by induction on n that each I_n satisfies the properties. For the base case, set I_0 clearly satisfies the property since it does not contain functional terms. For the inductive step, assume that some I_n satisfies the property, and consider an arbitrary term w_2, role R, and concept $A \in N_C \cup \{\top\}$. By the construction of $\Xi(\mathcal{K})$, there are only two types of clauses that may introduce new functional terms in $\mathsf{dom}(I_{n+1})$. First, such a term may be introduced by clauses of type 6 (see Table 1), but then the term clearly satisfies the required property. Second, a clause of the form $x \approx y \to f_{R,A}(x) \approx f_{R,A}(y)$ may be applied $w_1 \approx w_2 \in I_n$ and derive $f_{R,A}(w_1) \approx f_{R,A}(w_2) \in I_{n+1}$. If $f_{R,A}(w_2) \in \mathsf{dom}(I_n)$, then set I_{n+1} satisfies the required property by the induction hypothesis. Otherwise, term $f_{R,A}(w_2)$ occurs in I_{n+1} only in equality assertions, so $f_{R,A}(w_2) \notin \mathsf{dom}(I_{n+1})$, and the property holds vacuously. □

Let $J_0, J_1, \ldots$ be the sequence used to construct the minimal Herbrand model J of $\mathsf{D}(\mathcal{K})$. We assume w.l.o.g. that each J_{n+1} is obtained from J_n by applying a single clause occurring in $\mathsf{D}(\mathcal{K})$, apart from the clause defining the symmetry of $\approx$ which is always applied so as to keep the relation $\approx$ in J_n symmetric. We next show that each J_n satisfies the following property.

Lemma 13. *For each $n \in \mathbb{N}$ and all terms u_1 and u_2, if $u_1 \approx u_2 \in J_n$ and $u_2 \in \mathsf{aux}_{J_n}$, then $u_1 = u_2$.*

Proof. We prove the claim by the induction on n. For the base case, J_0 satisfies the property since aux_{J_0} is empty. For the inductive step, assume that some J_n satisfies the property; we show that the same holds for J_{n+1}. We consider the various clauses that may derive an equality in J_{n+1}. The facts derived by a clause of the form $A(x) \to x \approx a$ vacuously satisfy the property since the derived fact involves terms that are not in $\mathsf{aux}_{J_{n+1}}$. Furthermore, a fact derived in J_{n+1} by applying either the reflexivity or the symmetry clause satisfies the property by the inductive hypothesis. We are left to consider the transitivity clause. Let u_1, u_2, and u_3 be arbitrary terms such that $\{u_1 \approx u_2,\ u_2 \approx u_3\} \subseteq J_n$, so the transitivity clause derives $u_1 \approx u_3 \in J_{n+1}$. We consider the interesting case in which $u_3 \in \mathsf{aux}_{J_{n+1}}$, so $u_3 \in \mathsf{aux}_{J_n}$. By the inductive hypothesis, we have $u_2 = u_3$; but then, $u_2 \in \mathsf{aux}_{J_n}$, and so again, by the inductive hypothesis, we have $u_1 = u_2$; finally, this implies that $u_1 = u_3$. □

Lemma 4. *Let I and J be the minimal Herbrand models of $\Xi(\mathcal{K})$ and $\mathsf{D}(\mathcal{K})$, respectively. Mapping δ satisfies the following five properties for all terms w_1 and w_2 in $\mathsf{dom}(I)$, each $B \in N_C \cup \{\top, \bot\}$, and each $R \in N_R$.*

1. *$B(\delta(w_1)) \in J$ implies that $B(w_1) \in I$.*

2. $R(\delta(w_1), \delta(w_2)) \in J$ *and* $\delta(w_2) \notin \mathsf{aux}_J$ *imply that* $R(w_1, w_2) \in I$.

3. $R(\delta(w_1), \delta(w_2)) \in J$ *and* $\delta(w_2) \in \mathsf{aux}_J$ *imply that* $\delta(w_2)$ *is of the form* $o_{P,A}$*, that* $R(w_1, f_{P,A}(w_1)) \in I$*, and that a term* w_1' *exists such that* $R(w_1', w_2) \in I$.

4. $\delta(w_1) \approx \delta(w_2) \in J$ *and* $\delta(w_2) \notin \mathsf{aux}_J$ *imply that* $w_1 \approx w_2 \in I$.

5. *For each term* u *occurring in* J*, term* $w \in \mathsf{dom}(I)$ *exists such that* $\delta(w) = u$.

Proof. Let $J_0, J_1, \ldots$ be the sequence as stated above. We prove the claim by induction on n.

Base case. Consider J_0. By definition, $\Xi(\mathcal{K}) \cup \Xi(\mathcal{K})_\approx$ and $\mathsf{D}(\mathcal{K}) \cup \mathsf{D}(\mathcal{K})_\approx$ contain the same facts, all of which only refer to the individuals in N_I and the predicates in $N_C \cup N_R \cup \{\top, \bot\}$. Since δ is the identity over N_I, aux_{J_0} is empty and $J_0 = I_0$, so properties 1–5 are satisfied.

Inductive step. Assume that some J_n satisfies properties 1–5; we show that the same holds for J_{n+1}. To this end, let w_1 and w_2 be arbitrary terms in $\mathsf{dom}(I)$. We next consider the various clauses in $\mathsf{D}(\mathcal{K}) \cup \mathsf{D}(\mathcal{K})_\approx$ that may derive fresh assertions in J_{n+1}.

- $A(x) \rightarrow B(x)$. Assume that $A(\delta(w_1)) \in J_n$, and so the clause derives $B(\delta(w_1)) \in J_{n+1}$. By the inductive hypothesis, we have $A(w_1) \in I$. Finally, since the same clause occurs in $\Xi(\mathcal{K})$, we have $B(w_1) \in I$.
- $A(x) \rightarrow x \approx a$. Assume that $A(\delta(w_1)) \in J_n$, and so for $\delta(w_2) = w_2 = a$ the clause derives $\delta(w_1) \approx \delta(w_2)$ in J_{n+1}. Clearly, we have $\delta(w_2) \notin \mathsf{aux}_{J_{n+1}}$. By the inductive hypothesis, we have $A(w_1) \in I$. Finally, since the same clause occurs in $\Xi(\mathcal{K})$, we have $w_1 \approx w_2 \in I$.
- $A_1(x) \wedge A_2(x) \rightarrow A(x)$. Assume that $A_1(\delta(w_1)) \in J_n$ and $A_2(\delta(w_1)) \in J_n$, and so the clause derives $A(\delta(w_1)) \in J_{n+1}$. By the inductive hypothesis, we have $\{A_1(w_1),\ A_2(w_1)\} \subseteq I$. Since the same clause occurs in $\Xi(\mathcal{K})$, we have $A(w_1) \in I$.
- $R(x, y) \wedge A_1(y) \rightarrow A(x)$. Assume that $R(\delta(w_1), \delta(w_2))$ and $A_1(\delta(w_2))$ are in contained J_n, and so the clause derives $A(\delta(w_1)) \in J_{n+1}$. We have the following two cases.
  - $\delta(w_2) \notin \mathsf{aux}_{J_n}$. By the inductive hypothesis, we then have $\{R(w_1, w_2),\ A_1(w_2)\} \subseteq I$.
  - $\delta(w_2) \in \mathsf{aux}_{J_n}$ and term $\delta(w_2)$ is an auxiliary individual of the form $o_{P,A}$. By the inductive hypothesis, we then have $\{R(w_1, f_{P,A}(w_1)),\ A_1(f_{P,A}(w_1))\} \subseteq I$.

  In either case, since the same clause occurs in $\Xi(\mathcal{K})$, we have $A(w_1) \in I$.
- $R(x, y) \rightarrow A(y)$. Assume that $R(\delta(w_1), \delta(w_2)) \in J_n$, so the clause derives $A(\delta(w_2)) \in J_{n+1}$. We have the following two cases.
  - $\delta(w_2) \notin \mathsf{aux}_{J_n}$. By the inductive hypothesis, we then have $R(w_1, w_2) \in I$.
  - $\delta(w_2) \in \mathsf{aux}_{J_n}$. By the inductive hypothesis, then there exists a term w_1' such that $R(w_1', w_2) \in I$.

  In either case, since the same clause occurs in $\Xi(\mathcal{K})$, we have $A(w_2) \in I$.
- $S(x, y) \rightarrow R(x, y)$. Assume that $S(\delta(w_1), \delta(w_2)) \in J_n$, and so the clause derives $R(\delta(w_1), \delta(w_2)) \in J_{n+1}$. We have the following two cases.
  - $\delta(w_2) \notin \mathsf{aux}_{J_n}$. By the inductive hypothesis, we have that $S(w_1, w_2) \in I$. Since the same clause occurs in $\Xi(\mathcal{K})$, we have $R(w_1, w_2) \in I$.
  - $\delta(w_2) \in \mathsf{aux}_{J_n}$ and $\delta(w_2)$ is an auxiliary individual of the form $o_{P,A}$. By the inductive hypothesis, then there exists a term w_1' such that $\{S(w_1, f_{P,A}(w_1)),\ S(w_1', w_2)\} \subseteq I$. Since the same clause occurs in $\Xi(\mathcal{K})$, we have that $\{R(w_1, f_{P,A}(w_1)),\ R(w_1', w_2)\} \subseteq I$.
- $A_1(x) \rightarrow R(x, o_{R,A})$. Assume that $A_1(\delta(w_1)) \in J_n$, so for $\delta(w_2) = o_{R,A}$ the clause derives $R(\delta(w_1), \delta(w_2))$ in J_{n+1}. By the inductive hypothesis, we then have $A(w_1) \in I$. Furthermore, by the definition of $\mathsf{D}(\mathcal{K})$, set $\Xi(\mathcal{K})$ contains the clause $A_1(x) \rightarrow R(x, f_{R,A}(x))$, so we have $R(w_1, f_{R,A}(w_1)) \in I$. We have the following cases.
  - $\delta(w_2) \notin \mathsf{aux}_{J_{n+1}}$. Thus, we also have $\delta(w_2) \notin \mathsf{aux}_{J_n}$, and so there exists some $c \in N_I$ such that $\delta(w_2) \approx \delta(c) \in J_n$ and $\delta(c) \notin \mathsf{aux}_{J_n}$. By the inductive hypothesis, we have $w_2 \approx c \in I$. Due to $\delta(w_2) = \delta(f_{R,A}(w_1))$ and the inductive hypothesis, we have $c \approx f_{R,A}(w_1) \in I$. Since $\approx$ is a congruence relation and $\{R(w_1, f_{R,A}(w_1)),\ c \approx f_{R,A}(w_1),\ c \approx w_2\} \subseteq I$, we have $R(w_1, w_2) \in I$, as required. By the inductive hypothesis, property 5 is also satisfied.
  - $\delta(w_2) \in \mathsf{aux}_{J_{n+1}}$. By the definition of δ, term w_2 is of the form $f_{R,A}(w_2')$, and, by the induction hypothesis, we have that $f_{R,A}(w_2') \in \mathsf{dom}(I)$. By Lemma 12, we have that $R(w_2', f_{R,A}(w_2')) \in I$. As stated above, $R(w_1, f_{R,A}(w_1)) \in I$, so property 3 is satisfied. Moreover, $\delta(f_{R,A}(w_1)) = o_{R,A}$, and so property 5 is satisfied as well.
- $A_1(x) \rightarrow A(o_{R,A})$. Assume that $A_1(\delta(w_1)) \in J_n$, so for $\delta(w_2) = o_{R,A}$ the clause derives $A(\delta(w_2)) \in J_{n+1}$. By the definition of δ, term w_2 is of the form $f_{R,A}(w_2')$. By Lemma 12 and $w_2 \in \mathsf{dom}(I)$, we have $A(w_2) \in I$.

- $\to x \approx x$. Assume that $\delta(w_1)$ occurs in J_n, so the clause derives $\delta(w_1) \approx \delta(w_2) \in J_{n+1}$ with $\delta(w_1) = \delta(w_2)$. We consider the interesting case when $\delta(w_2) \notin \mathsf{aux}_{J_{n+1}}$, and so $\delta(w_2) \notin \mathsf{aux}_{J_n}$. Then, an individual $c \in N_I$ exists such that $\{\delta(w_1) \approx c,\ c \approx \delta(w_2)\} \subseteq J_n$. By the inductive hypothesis, we have that $\{w_1 \approx c,\ c \approx w_2\} \subseteq I$. By the transitivity of $\approx$, we have $w_1 \approx w_2 \in I$.
- $x_1 \approx x_2 \to x_2 \approx x_1$. Assume that $\delta(w_1) \approx \delta(w_2) \in J_n$, so the clause derives $\delta(w_2) \approx \delta(w_1) \in J_{n+1}$. We consider the interesting case when $\delta(w_1) \notin \mathsf{aux}_{J_{n+1}}$; clearly, we have $\delta(w_1) \notin \mathsf{aux}_{J_n}$ as well. Since predicate $\approx$ is symmetric in J_n, we have $\delta(w_2) \approx \delta(w_1) \in J_n$. By the inductive hypothesis, we have $w_2 \approx w_1 \in I$.
- $x_1 \approx x_3 \wedge x_3 \approx x_2 \to x_1 \approx x_2$. Assume that set J_n contains $\delta(w_1) \approx \delta(w_3)$ and $\delta(w_3) \approx \delta(w_2)$, so the clause derives $\delta(w_1) \approx \delta(w_2) \in J_{n+1}$. The only interesting case is when $\delta(w_2) \notin \mathsf{aux}_{J_{n+1}}$; clearly, then $\delta(w_2) \notin \mathsf{aux}_{J_n}$. By Lemma 13, then $\delta(w_3) \notin \mathsf{aux}_{J_n}$. Finally, by the inductive hypothesis, then $\{w_1 \approx w_3,\ w_3 \approx w_2\} \subseteq I$, which implies $w_1 \approx w_2 \in I$.
- $A(x) \wedge x \approx y \to A(y)$. Assume that facts $A(\delta(w_1))$ and $\delta(w_1) \approx \delta(w_2)$ are contained in J_n, so the clause derives $A(\delta(w_2)) \in J_{n+1}$. By the inductive hypothesis, we have $A(w_1) \in I$. We consider the following two cases.
  - $\delta(w_2) \notin \mathsf{aux}_{J_n}$. By the inductive hypothesis, we have $w_1 \approx w_2 \in I$, and so $A(w_2) \in I$.
  - $\delta(w_2) \in \mathsf{aux}_{J_n}$. By Lemma 13, then $\delta(w_1) = \delta(w_2)$, so $A(\delta(w_2)) \in J_n$. Finally, by the inductive hypothesis, we then have $A(w_2) \in I$.
- $R(x, y) \wedge x \approx z \to R(z, y)$. Assume that set J_n contains $R(\delta(w_1), \delta(w_2))$ and $\delta(w_1) \approx \delta(w_3)$, so the clause derives $R(\delta(w_3), \delta(w_2)) \in J_{n+1}$. We consider the following two cases.
  - $\delta(w_2) \notin \mathsf{aux}_{J_n}$. By the inductive hypothesis, we have $R(w_1, w_2) \in I$. We distinguish two additional cases. First, assume that $\delta(w_3) \in \mathsf{aux}_{J_n}$. By Lemma 13, we have $\delta(w_1) = \delta(w_3)$, and so $R(\delta(w_3), \delta(w_2)) \in J_n$. By the inductive hypothesis, then $R(w_3, w_2) \in I$. Second, assume that $\delta(w_3) \notin \mathsf{aux}_{J_n}$. By the inductive hypothesis, we have $w_1 \approx w_3 \in I$, and so we have $R(w_3, w_2) \in I$ as well.
  - $\delta(w_2) \in \mathsf{aux}_{J_n}$ and $\delta(w_2) = o_{P,A}$. By the inductive hypothesis, some w'_1 exists s.t. $\{R(w_1, f_{P,A}(w_1)),\ R(w'_1, w_2)\} \subseteq I$. We distinguish two additional cases. First, assume that $\delta(w_3) \in \mathsf{aux}_{J_n}$. By Lemma 13, we have $\delta(w_1) = \delta(w_3)$, which further implies $R(\delta(w_3), \delta(w_2)) \in J_n$. By the inductive hypothesis, then we have $\{R(w_3, f_{P,A}(w_3)),\ R(w'_1, w_2)\} \subseteq I$. Second, assume that $\delta(w_3) \notin \mathsf{aux}_{J_n}$. By the inductive hypothesis, we have $w_1 \approx w_3 \in I$. By the functional reflexivity clauses, then $f_{R,B}(w_1) \approx f_{R,B}(w_3) \in I$, which again implies $\{R(w_3, f_{R,B}(w_3)),\ R(w'_1, w_2)\} \subseteq I$.
- $R(x, y) \wedge y \approx z \to R(x, z)$. Assume that set J_n contains $R(\delta(w_1), \delta(w_2))$ and $\delta(w_2) \approx \delta(w_3)$, so the clause derives $R(\delta(w_1), \delta(w_3)) \in J_{n+1}$. We consider the following two cases.
  - $\delta(w_2) \notin \mathsf{aux}_{J_n}$. By Lemma 13, then $\delta(w_3) \notin \mathsf{aux}_{J_n}$, and so $\delta(w_3) \notin \mathsf{aux}_{J_{n+1}}$ as well. By the inductive hypotheses, then $R(w_1, w_2)$ and $w_2 \approx w_3$ are in I, so $R(w_1, w_3) \in I$ as well.
  - $\delta(w_2) \in \mathsf{aux}_{J_n}$ and $\delta(w_2)$ is of the form $o_{P,A}$. By Lemma 13, then $\delta(w_3) = \delta(w_2)$, which implies $R(\delta(w_1), \delta(w_3)) \in J_n$. Finally, by the inductive hypothesis, then there exists a term w'_1 such that $\{R(w_1, f_{R,B}(w_1)),\ R(w'_1, w_3)\} \subseteq I$. ☐

Proof of Proposition 5

Proposition 5. *For $\mathcal{K}$ an arbitrary $\mathcal{ELHO}^r_\bot$ knowledge base, $\Xi(\mathcal{K}) \models \exists y.\bot(y)$ if and only if $\mathsf{D}(\mathcal{K}) \models \exists y.\bot(y)$. Furthermore, the satisfiability of $\mathcal{K}$ can be checked in time polynomial in the size of $\mathcal{K}$.*

Proof. From Lemmas 3 and 4, we have $\bot(w) \in I$ if and only if $\bot(\delta(w)) \in J$. Thus, $\mathcal{K}$ is unsatisfiable if and only if individual u exists such that $\mathsf{D}(\mathcal{K}) \models \bot(u)$. Furthermore, to check the latter, we can compute J and check whether an individual u exists such that $\bot(u) \notin J$. Since the number of variables occurring in each datalog clause is bounded by a constant, the computation of J can be implemented in polynomial time in the size of $\mathcal{K}$ (Dantsin et al. 2001). ☐

Proof of Theorem 10

We first show that the minimal Herbrand model I of $\Xi(\mathcal{K})$ resembles a forest structure. Let $I_0, I_1, \ldots$ be the sets used to generate I; for simplicity, in the rest of this section we assume w.l.o.g. that the clauses are applied in a way so that relation $\approx$ is symmetric in each I_n. Furthermore, for each term w, we define the *size of* w as follows.

$$|w| = \begin{cases} 0 & \text{if } w \in N_I, \\ 1 + |w'| & \text{if } w \text{ is of the form } f_{T,A}(w'). \end{cases}$$

Finally, we define the *depth of w in I* as follows.

$$\mathsf{d}(w, I) = \begin{cases} 0 & \text{if } w \notin \mathsf{aux}_I, \\ 1 + \mathsf{d}(w', I) & \text{if } w \in \mathsf{aux}_I \text{ and } w = f_{T,A}(w'). \end{cases}$$

Lemma 14. *Interpretation I satisfies the following three properties for all terms w_1, w'_1, w_2, and w'_2, all roles R, S, and T, and each concept $A \in N_C \cup \{\top\}$.*

P1. $R(w_1', f_{T,A}(w_1)) \in I$, $S(w_2', f_{T,A}(w_2)) \in I$,
$f_{T,A}(w_1) \approx f_{T,A}(w_2) \in I$, *and* $f_{T,A}(w_2) \in \mathsf{aux}_I$ *imply* $w_1' \approx w_2' \in I$.

P2. $w_1 \approx w_2 \in I$ *implies* $\mathsf{d}(w_1, I) = \mathsf{d}(w_2, I)$.

P3. $R(w_1', f_{T,A}(w_1)) \in I$ *and* $f_{T,A}(w_1) \in \mathsf{aux}_I$ *imply that* $\mathsf{d}(f_{T,A}(w_1), I) = 1 + \mathsf{d}(w_1', I)$.

Proof. To prove properties P1–P3, we first show by induction on n that each I_n satisfies the following two auxiliary properties for all terms w', w, w_1, and w_2, all roles R, T, and T', and all concepts A and A' in $N_C \cup \{\top\}$.

A1. $f_{T',A'}(w_2) \in \mathsf{aux}_{I_n}$ and $f_{T,A}(w_1) \approx f_{T',A'}(w_2) \in I_n$ imply that $T = T'$, $A = A'$, and $w_1 \approx w_2 \in I$.

A2. $f_{T,A}(w) \in \mathsf{aux}_{I_n}$ and $R(w', f_{T,A}(w)) \in I_n$ imply that a term w'' exists such that I contains $T(w'', f_{T,A}(w''))$, $w' \approx w''$, and $f_{T,A}(w) \approx f_{T,A}(w'')$.

Base case. By definition, I_0 does not contain functional terms, so properties A1 and A2 are vacuously true.

Inductive step. Assume that I_n satisfies properties A1 and A2; we show that the same holds for I_{n+1} by considering in turn the various clauses that may introduce fresh assertions into I_{n+1}. We consider only the interesting cases in where an equality or a binary assertion is derived, since all other clauses trivially preserve A1 and A2. Let w', w, w_1, w_1', and w_2 be arbitrary terms, let R, T, and T' be arbitrary roles, and let A and A' be arbitrary concepts in $N_C \cup \{\top\}$.

- $A_1(x) \to x \approx a$. Assume that $A_1(w_1) \in I_n$, so the clause derives $w_1 \approx a \in I_{n+1}$. Since $a \notin \mathsf{aux}_{I_{n+1}}$, properties A1 and A2 are preserved.
- $\to x \approx x$. Assume that $f_{T,A}(w_1)$ occurs in I_n, so the clause derives $f_{T,A}(w_1) \approx f_{T,A}(w_1) \in I_{n+1}$; the interesting case is when $f_{T,A}(w_1) \in \mathsf{aux}_{I_{n+1}}$. Since $f_{T,A}(w_1)$ occurs in I_n, then w_1 occurs in the Herbrand universe of $\Xi(\mathcal{K})$. By reflexivity, then $w_1 \approx w_1 \in I$, as required for A1. Furthermore, this derivation clearly preserves A2.
- $x \approx y \to f_{T,A}(x) \approx f_{T,A}(y)$. Assume $w_1 \approx w_2 \in I_n$, so the clause derives $f_{T,A}(w_1) \approx f_{T,A}(w_2) \in I_{n+1}$; the interesting case is when $f_{T,A}(w_2) \in \mathsf{aux}_{I_{n+1}}$. By assumption, $w_1 \approx w_2 \in I_n$, and so $w_1 \approx w_2 \in I$, as required for property A1. Furthermore, this derivation clearly preserves A2.
- $x_1 \approx x_2 \to x_2 \approx x_1$. Assume that $f_{T',A'}(w_2) \approx f_{T,A}(w_1) \in I_n$, so the clause derives $f_{T,A}(w_1) \approx f_{T',A'}(w_2) \in I_{n+1}$; the interesting case is when $f_{T',A'}(w_2) \in \mathsf{aux}_{I_{n+1}}$, which clearly implies $f_{T',A'}(w_2) \in \mathsf{aux}_{I_n}$. Since relation $\approx$ is symmetric in I_n, we have $f_{T,A}(w_1) \approx f_{T',A'}(w_2) \in I_n$; but then, by the inductive hypothesis, we have $T = T'$, $A = A'$, and $w_1 \approx w_2 \in I$, as required for property A1. Furthermore, this derivation clearly preserves A2.
- $x_1 \approx x_2 \wedge x_2 \approx x_3 \to x_1 \approx x_3$. Assume that I_n contains $f_{T,A}(w_1) \approx f_{T',A'}(w_1')$ and $f_{T',A'}(w_1') \approx f_{T'',A''}(w_2)$, so the clause derives $f_{T,A}(w_1) \approx f_{T'',A''}(w_2) \in I_{n+1}$; the interesting case is when $f_{T'',A''}(w_2) \in \mathsf{aux}_{I_{n+1}}$, which clearly implies $f_{T'',A''}(w_2) \in \mathsf{aux}_{I_n}$. Clearly, we then also have $f_{T',A'}(w_1') \in \mathsf{aux}_{I_n}$. By the inductive hypothesis, we have $T = T' = T''$, $A = A' = A''$, and $\{w_1 \approx w_1',\ w_1' \approx w_2\} \subseteq I$. Thus, we have $w_1 \approx w_2 \in I$, as required for property A1. Furthermore, this derivation clearly preserves A2.
- $A_1(x) \to T(x, f_{T,A}(x))$. Assume that $A_1(w') \in I_n$, so the clause derives $T(w', f_{T,A}(w')) \in I_{n+1}$; the interesting case is when $f_{T,A}(w') \in \mathsf{aux}_{I_{n+1}}$ and $w' = w$. Then, for $w'' = w = w'$, we have
$$\{T(w'', f_{T,A}(w'')),\ w'' \approx w'',\ f_{T,A}(w'') \approx f_{T,A}(w'')\} \subseteq I,$$
as required for property A2. Furthermore, this derivation clearly preserves A1.
- $P(x, y) \to R(x, y)$. Assume that $P(w', f_{T,A}(w)) \in I_n$, so the clause derives $R(w', f_{T,A}(w)) \in I_{n+1}$; the interesting case is when $f_{T,A}(w) \in \mathsf{aux}_{I_{n+1}}$, which implies $f_{T,A}(w) \in \mathsf{aux}_{I_n}$. By the inductive hypothesis, then a term w'' exists such that $\{T(w'', f_{T,A}(w'')),\ w' \approx w'',\ f_{T,A}(w) \approx f_{T,A}(w'')\} \subseteq I$, as required for property A2. Furthermore, this derivation clearly preserves A1.
- $R(x, y) \wedge x \approx z \to R(z, y)$. Assume that $\{R(w_1', f_{T,A}(w)),\ w_1' \approx w'\} \subseteq I_n$, so the clause derives $R(w', f_{T,A}(w)) \in I_{n+1}$; the interesting case is when $f_{T,A}(w) \in \mathsf{aux}_{I_{n+1}}$, which implies $f_{T,A}(w) \in \mathsf{aux}_{I_n}$. By the inductive hypothesis, a term w'' exists such that $\{T(w'', f_{T,A}(w'')),\ w_1' \approx w'',\ f_{T,A}(w) \approx f_{T,A}(w'')\} \subseteq I$. By the transitivity of $\approx$, we have $w' \approx w'' \in I$, as required for property A2. Furthermore, this derivation clearly preserves A1.
- $R(x, y) \wedge y \approx z \to R(x, z)$. Assume that $\{R(w', f_{T,A}(w_1)),\ f_{T,A}(w_1) \approx f_{T,A}(w)\} \subseteq I_n$, and so the clause derives the fact $R(w', f_{T,A}(w)) \in I_{n+1}$; the interesting case is when $f_{T,A}(w) \in \mathsf{aux}_{I_{n+1}}$, which implies $f_{T,A}(w) \in \mathsf{aux}_{I_n}$. Then, clearly $f_{T,A}(w_1) \in \mathsf{aux}_{I_n}$. By the inductive hypothesis, a term w'' exists such that
$$\{T(w'', f_{T,A}(w'')),\ w' \approx w'',\ f_{T,A}(w_1) \approx f_{T,A}(w'')\} \subseteq I.$$
By the transitivity of $\approx$, then $f_{T,A}(w) \approx f_{T,A}(w'') \in I$, as required for property A2. Furthermore, this derivation clearly preserves A1.

We are now ready to show properties P1–P3.

PROPERTY P1. Let w'_1, w_1, w'_2, w_2 be arbitrary terms, let R, S, and T be arbitrary roles, and let A be an arbitrary concept in $N_C \cup \{\top\}$. Assume that $\{R(w'_1, f_{T,A}(w_1)),\ S(w'_2, f_{T,A}(w_2)),\ f_{T,A}(w_1) \approx f_{T,A}(w_2)\} \subseteq I$ and $f_{T,A}(w_2) \in \mathsf{aux}_I$. By applying property A2 to $R(w'_1, f_{T,A}(w_1))$ and $S(w'_2, f_{T,A}(w_2))$, we have that two terms w''_1 and w''_2 exist such that

$$\{T(w''_1, f_{T,A}(w''_1)),\ w'_1 \approx w''_1, f_{T,A}(w_1) \approx f_{T,A}(w''_1)\} \subseteq I, \text{ and}$$
$$\{T(w''_2, f_{T,A}(w''_2)),\ w'_2 \approx w''_2, f_{T,A}(w_2) \approx f_{T,A}(w''_2)\} \subseteq I.$$

By the transitivity of $\approx$, we have that $f_{T,A}(w''_1) \approx f_{T,A}(w''_2) \in I$, and so by Property A1, we conclude that $w''_1 \approx w''_2 \in I$. Finally, since $\{w'_1 \approx w''_1,\ w''_1 \approx w''_2,\ w''_2 \approx w'_2\} \subseteq I$, by the transitivity of $\approx$, we get $w'_1 \approx w'_2 \in I$, as required.

PROPERTY P2. We show by induction on $n \in \mathbb{N}$ that, for all terms w_1 and w_2 such that $|w_1| \leq |w_2| \leq n$, if $w_1 \approx w_2 \in I$, then $\mathsf{d}(w_1, I) = \mathsf{d}(w_2, I)$.

Base case. Let w_1 and w_2 be arbitrary terms such that $|w_1| = |w_2| = 0$ and $w_1 \approx w_2 \in I$. By the definition of $|\cdot|$, then $\{w_1, w_2\} \subseteq N_I$, so $\mathsf{d}(w_1, I) = \mathsf{d}(w_2, I) = 0$.

Inductive step. Consider an arbitrary $n \in \mathbb{N}$ and assume that the required property holds for all terms w'_1 and w'_2 such that $|w'_1| \leq |w'_2| \leq n$; we show that the same holds for arbitrary terms w_1 and w_2 such that $|w_1| \leq |w_2| \leq n+1$. We consider the interesting case when $w_1 \approx w_2 \in I$, for which we consider two cases. First, if $w_2 \notin \mathsf{aux}_I$, then $\mathsf{d}(w_1, I) = \mathsf{d}(w_2, I) = 0$. Second, if $w_2 \in \mathsf{aux}_I$, then by property A1 there exist two terms w'_1 and w'_2, a role T, and a concept $A \in N_C \cup \{\top\}$ such that w_1 is of the form $f_{T,A}(w'_1)$, term w_2 is of the form $f_{T,A}(w'_2)$, and $w'_1 \approx w'_2 \in I$. By the inductive hypothesis, then $\mathsf{d}(w'_1, I) = \mathsf{d}(w'_2, I)$. Finally, by definition, we have $\mathsf{d}(w_1, I) = \mathsf{d}(w_2, I) = 1 + \mathsf{d}(w'_2, I)$, as required.

PROPERTY P3. Let w'_1 and w_1 be arbitrary terms, let R and T be arbitrary roles, let $A \in N_C \cup \{\top\}$ be an arbitrary concept, and assume that $R(w'_1, f_{T,A}(w_1)) \in I$ and $f_{T,A}(w_1) \in \mathsf{aux}_I$. By property A2, then there exists a term w''_1 such that $\{T(w''_1, f_{T,A}(w''_1)),\ w'_1 \approx w''_1,\ f_{T,A}(w_1) \approx f_{T,A}(w''_1)\} \subseteq I$. By the definition of $\mathsf{d}(\cdot)$, then $\mathsf{d}(f_{T,A}(w''_1), I) = 1 + \mathsf{d}(w''_1, I)$. Furthermore, by property P2, then $\mathsf{d}(f_{T,A}(w_1), I) = \mathsf{d}(f_{T,A}(w''_1), I)$ and $\mathsf{d}(w'_1, I) = \mathsf{d}(w''_1, I)$. Finally, these observations imply that $\mathsf{d}(f_{T,A}(w_1), I) = 1 + \mathsf{d}(w'_1, I)$, as required. □

We now have all the ingredients required to prove Theorem 10. We start by showing completeness.

Lemma 15 (Completeness)**.** *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a satisfiable $\mathcal{ELHO}^r_\bot$ KB, let $q = \exists \vec{y}.\psi(\vec{x}, \vec{y})$ be a CQ, and let $\pi : \vec{x} \mapsto N_I$ be a candidate answer for q. Then, $\Xi(\mathcal{K}) \models \pi(q)$ implies that a substitution τ exists such* $\mathsf{dom}(\tau) = \vec{x} \cup \vec{y}$, $\tau|_{\vec{x}} = \pi$, $\mathsf{D}(\mathcal{K}) \models \tau(q)$, *and* $\mathsf{isSpur}(q, \mathsf{D}(\mathcal{K}), \tau) = \mathsf{f}$.

Proof. Let I and J be the minimal Herbrand models of $\Xi(\mathcal{K})$ and $\mathsf{D}(\mathcal{K})$, respectively. Since $\Xi(\mathcal{K}) \models \pi(q)$, a substitution π^* exists such that $\mathsf{dom}(\pi^*) = \vec{x} \cup \vec{y}$, $\pi^*|_{\vec{x}} = \pi$, and $\pi^*(q) \subseteq I$. Let δ be the mapping from I to J defined in the section about the datalog rewriting of $\mathcal{K}$. We define τ as the substitution such that, for each term $t \in N_T(q)$, we have $\tau(t) := \delta(\pi^*(t))$. Finally, let $\sim$ be the relation for τ, q, and $\mathsf{D}(\mathcal{K})$ as specified in Definition 7. Since δ is a homomorphism from I to J by Lemma 3, we have $J \models \tau(q)$. We next prove $\mathsf{isSpur}(q, \tau, \mathsf{D}(\mathcal{K})) = \mathsf{f}$ by showing that all conditions of Definition 9 are satisfied.

(Condition a) By the definition of τ, for each $x \in \vec{x}$, we have $\tau(x) \in N_I$.

(Condition b) We prove that, for each $s \sim t$, we have $\tau(s) \approx \tau(t) \in J$ and $\pi^*(s) \approx \pi^*(t) \in I$. We proceed by induction on the number of steps required to derive $s \sim t$. For the base case, the empty relation $\sim$ clearly satisfies the two properties. For the inductive step, consider an arbitrary relation $\sim$ obtained in n steps that satisfies these constraints; we show that the same holds for all constraints derivable from $\sim$. Since relation $\approx$ in both J and I is reflexive, symmetric, and transitive, the derivation of $s \sim t$ due to reflexivity, symmetry, or transitivity clearly preserves the required properties; thus, we focus on the (fork) rule. Let s', s, t', and t be arbitrary terms in $N_T(q)$, and let R and P be arbitrary roles such that $s' \sim t'$ is obtained in n steps, atoms $R(s, s')$ and $P(t, t')$ occur in q, and $\tau(s') \in \mathsf{aux}_{\mathsf{D}(\mathcal{K})}$. By the inductive hypothesis, we have $\tau(s') \approx \tau(t') \in J$ and $\pi^*(s') \approx \pi^*(t') \in I$. Since J is the minimal Herbrand model of $\mathsf{D}(\mathcal{K})$, we have $\tau(t') \in \mathsf{aux}_J$, so no individual $c \in N_I$ exists such that $\tau(t') \approx c \in J$. By Lemmas 3 and 4, $\tau(t') \notin \mathsf{aux}_J$ if and only if $\pi^*(t') \notin \mathsf{aux}_I$; hence, $\pi^*(t') \in \mathsf{aux}_I$. Since $\{R(\pi^*(s), \pi^*(s')),\ P(\pi^*(t), \pi^*(t')),\ \pi^*(s') \approx \pi^*(t')\} \subseteq I$ and $\pi^*(t') \in \mathsf{aux}_I$, by property P1 of Lemma 14 we have $\pi^*(s) \approx \pi^*(t) \in I$. Finally, since δ is a homomorphism (see Lemma 3), by the construction of τ we have $\tau(s) \approx \tau(t) \in J$, as required.

(Condition c) To show that q is aux-acyclic w.r.t. τ and $\mathsf{D}(\mathcal{K})$, we assume the opposite; hence, there exists a sequence of vertices $v_0, \dots, v_m \in V_{\mathsf{aux}}$ such that $m > 0$, for each $0 \leq i < m$ we have $\langle v_i, v_{i+1} \rangle \in E_{\mathsf{aux}}$, and $v_m = v_0$. Consider an arbitrary $i \leq m$ and the corresponding edge $\langle v_i, v_{i+1} \rangle \in E_{\mathsf{aux}}$. By the definition of E_{aux}, an atom $R_i(s_i, s_{i+1})$ exists in q such that $\gamma(s_i) = v_i$ and $\gamma(s_{i+1}) = v_{i+1}$; hence, we have $s_i \sim v_i$ and $s_{i+1} \sim v_{i+1}$. Since τ satisfies all the constraints in $\sim$, by the definition of G_{aux} we have that $\{\tau(s_i),\ \tau(s_{i+1})\} \subseteq \mathsf{aux}_{\mathsf{D}(\mathcal{K})}$. By Lemmas 3 and 4, then $\{\pi^*(s_i),\ \pi^*(s_{i+1})\} \subseteq \mathsf{aux}_I$ as well. In addition, since $R_i(s_i, s_{i+1})$ is an atom in q, we have $R_i(\pi^*(s_i), \pi^*(s_{i+1})) \in I$. Also, since $s_i \sim v_i$, $s_{i+1} \sim v_{i+1}$, and substitution π^* satisfies the constraints in $\sim$, we have $\{\pi^*(s_i) \approx \pi^*(v_i),\ \pi^*(s_{i+1}) \approx \pi^*(v_{i+1})\} \subseteq I$. Then, by property P2 of Lemma 14, we have

$$\begin{aligned} \mathsf{d}(\pi^*(s_i), I) &= \mathsf{d}(\pi^*(v_i), I) \qquad \text{and} \\ \mathsf{d}(\pi^*(s_{i+1}), I) &= \mathsf{d}(\pi^*(v_{i+1}), I). \end{aligned}$$

Finally, since $R(\pi^*(s_i), \pi^*(s_{i+1})) \in I$, by property P3 of Lemma 14 we have $\mathsf{d}(\pi^*(v_{i+1}), I) = 1 + \mathsf{d}(\pi^*(v_i), I)$. But then, since $v_m = v_0$, we also have $\mathsf{d}(\pi^*(v_m), I) = \mathsf{d}(\pi^*(v_0), I)$, which is a contradiction. □

We are left to prove the soundness of our approach. Let τ be an arbitrary substitution for q w.r.t. $\mathsf{D}(\mathcal{K})$ such that $J \models \tau(q)$ and $\mathsf{isSpur}(q, \tau, \mathsf{D}(\mathcal{K})) = \mathsf{f}$. Furthermore, let G_{aux} be the graph as specified in Definition 8. We next show that $I \models \tau|_{\vec{x}}(q)$. In order to do so, we first define the graph G_q of the query q.

Definition 16. *Let γ and V_{aux} be as in Definition 8. The* query graph *$G_q = \langle V_q, E_q\rangle$ is the directed graph defined as follows.*

- *V_q is the smallest set containing $\gamma(t)$ for each $t \in N_T(q)$.*
- *E_q is the smallest set containing $\langle \gamma(s), \gamma(t)\rangle$ for all terms $\{s, t\} \subseteq N_T(q)$ such that query q contains $R(s, t)$ for some R.*

Vertex $v \in V_q$ is a root *if $v \notin V_{\mathsf{aux}}$ or, for each vertex $v' \in V_q$, we have $\langle v', v\rangle \notin E_q$.*

Clearly, by the definition, G_{aux} is a subgraph G_q. We prove the soundness claim in three steps. First, we show that the graph G_q is a forest. Second, we define by structural induction on the forest G_q a substitution π for q w.r.t. $\Xi(\mathcal{K})$ such that $\tau|_{\vec{x}} = \pi|_{\vec{x}}$. Third, we prove that $I \models \pi(q)$ holds.

Lemma 17. *If $\mathsf{isSpur}(q, \tau, \mathsf{D}(\mathcal{K})) = \mathsf{f}$, then G_q is a forest.*

Proof. Due to $\mathsf{isSpur}(q, \tau, \mathsf{D}(\mathcal{K})) = \mathsf{f}$, we have that G_{aux} is a direct acyclic graph. Consider an arbitrary vertex $v \in V_{\mathsf{aux}}$ and arbitrary vertices $v_1, v_2 \in V_q$ such that $\{\langle v_1, v\rangle, \langle v_2, v\rangle\} \subseteq E_q$; we next show that $v_1 = v_2$. By the definition of G_q, we have that $\{s, s', t, t'\} \subseteq N_T(q)$, and that roles R and P exist such that all of the following conditions are satisfied:

- atoms $R(s, s')$ and $P(t, t')$ are in q;
- $\gamma(s') = v = \gamma(t')$, $\gamma(s) = v_1$, and $\gamma(t) = v_2$; and,
- $\{\tau(s'), \tau(t')\} \subseteq \mathsf{aux}_J$.

Due to the (fork) rule, we have $s \sim t$. By the definition of γ, we have $\gamma(s) = \gamma(t)$, which implies $v_1 = v_2$, as required. □

By structural induction on the forest-shaped graph G_q, we next define the substitution π as follows; we will later show that $\Xi(\mathcal{K}) \models \pi(q)$.

- For the base case, let v be an arbitrary root of G_q. For each term $t \in N_T(q)$ such that $\gamma(t) = v$, we define $\pi(t)$ as an arbitrary term $w \in \mathsf{dom}(I)$ such that $\delta(w) = \tau(t)$.
- For the inductive step, let v be an arbitrary vertex of G_q such that $v \in V_{\mathsf{aux}}$, term $\tau(v)$ is of the form $o_{R,A}$, the value of $\pi(v)$ is undefined, v' is the unique vertex of G_q such that $\langle v', v\rangle \in E_q$, and $\pi(v')$ has already been defined. For each term $t \in N_T(q)$ such that $\gamma(t) = v$, we define $\pi(t) := f_{R,A}(\pi(v'))$.

Lemma 18. *Substitution π satisfies the two following properties for each term $v \in V_q$ and all terms $s, t \in N_T(q)$ such that $\gamma(s) = v = \gamma(t)$:*

M1. $\delta(\pi(s)) = \tau(s)$*, and*

M2. $\pi(s) \approx \pi(t) \in I$*.*

Proof. We prove properties M1 and M2 by the structural induction on the forest G_q.

Base case. Let v be an arbitrary root of G_q, and let $s, t \in N_T(q)$ be arbitrary terms such that $\gamma(s) = v = \gamma(t)$. Property M1 follows from the fact that $\pi(s) \in \{w \in \mathsf{dom}(I) \mid \delta(w) = \tau(s)\}$. We next prove property M2. By the definition of γ, we have that $s \sim t$. Since $\mathsf{isSpur}(q, \tau, \mathsf{D}(\mathcal{K})) = \mathsf{f}$, we have $\tau(s) \approx \tau(t) \in J$. We have the following two cases.

- Assume that $v \in V_{\mathsf{aux}}$. Clearly, $\{\tau(s),\ \tau(t)\} \subseteq \mathsf{aux}_J$. By the construction of J, there exists $n \in \mathbb{N}$ such that $\tau(s) \approx \tau(t) \in J_n$ and $\tau(t) \in \mathsf{aux}_{J_n}$. By Lemma 13, we have $\tau(s) = \tau(t)$. Thus, $\pi(s) = \pi(t)$ and $\pi(s) \approx \pi(t) \in I$, as required.
- Assume that $v \notin V_{\mathsf{aux}}$. Then, we have $\tau(t) \notin \mathsf{aux}_J$ and, by Lemma 4, we have $\pi(s) \approx \pi(t) \in I$.

Inductive step. Let $v \in V_{\mathsf{aux}}$ be an arbitrary vertex, let $s, t \in N_T(q)$ be arbitrary terms such that $\gamma(s) = v = \gamma(t)$, and assume that $\tau(v)$ is of the form $o_{R,A}$. By the definition of γ, we have that $s \sim t$. Since $\mathsf{isSpur}(q, \tau, \mathsf{D}(\mathcal{K})) = \mathsf{f}$, we have $\tau(s) \approx \tau(t) \in J$. Since $v \in V_{\mathsf{aux}}$, we have $\{\tau(v), \tau(s), \tau(t)\} \subseteq \mathsf{aux}_J$. Then, by the construction of J, some $n \in \mathbb{N}$ exists such that $\{\tau(v) \approx \tau(s),\ \tau(s) \approx \tau(t)\} \subseteq J_n$ and $\{\tau(s), \tau(t)\} \subseteq \mathsf{aux}_{J_n}$. By Lemma 13, we have $\tau(v) = \tau(s)$ and $\tau(s) = \tau(t)$. Now let v' be the unique vertex of G_q such that $\langle v', v\rangle \in E$. By definition, $\pi(s) = f_{R,A}(\pi(v')) = \pi(t)$. Also, by the definition of δ, we have $\delta(f_{R,A}(\pi(v'))) = o_{R,A}$, so property M1 holds. By the reflexivity of $\approx$, we have $\pi(s) \approx \pi(t) \in I$, and so property M2 holds, as required. □

We finally prove the soundness of our approach.

Lemma 19 (Soundness). *Let I and J be the minimal Herbrand models of $\Xi(\mathcal{K})$ and $\mathsf{D}(\mathcal{K})$, respectively; let $q = \exists \vec{y}.\psi(\vec{x}, \vec{y})$ be an arbitrary CQ; and let τ be an arbitrary substitution such that $\tau(q) \subseteq J$ and $\mathsf{isSpur}(q, \tau, \mathsf{D}(\mathcal{K})) = \mathsf{f}$. Then, $\tau|_{\vec{x}}(q) \in I$.*

Proof. For q and τ as specified in the lemma, let π be the substitution defined as specified just before Lemma 18, and assume that $\mathsf{isSpur}(q, \mathsf{D}(\mathcal{K}), \tau) = \mathsf{f}$. By definition, we have $\pi|_{\vec{x}} = \tau|_{\vec{x}}$. We next show that $\pi(q) \subseteq I$.

First, let $A(t)$ be an arbitrary unary atom of q, we show that $A(\pi(t)) \in I$. By assumption, we have $A(\tau(t)) \in J$. By Lemma 4, for each term $w \in \mathsf{dom}(I)$ such that $\delta(w) = \tau(t)$, we have that $A(w) \in I$. By property M1 of Lemma 18, we have $A(\pi(t)) \in I$.

Second, let $R(t', t)$ be an arbitrary atom of q; we show that $R(\pi(t'), \pi(t)) \in I$. By assumption, we have $R(\tau(t'), \tau(t)) \in J$. We distinguish the following two cases.

1. Assume that $\tau(t) \notin \mathsf{aux}_J$. By Lemma 4, for all terms $w', w \in \mathsf{dom}(I)$ such that $\delta(w') = \tau(t')$ and $\delta(w) = \tau(t)$, we have $R(w', w) \in I$. By property M1 of Lemma 18, we have $R(\pi(t'), \pi(t)) \in I$.
2. Assume that $\tau(t) \in \mathsf{aux}_J$, and assume that $\tau(t)$ is of the form $o_{R,A}$. Furthermore, let v' be the unique vertex of G_q such that $\langle v', \gamma(t) \rangle \in E_q$ and $\gamma(t') = v'$. By the definition of π, we have $\pi(t) = f_{R,A}(\pi(v'))$. Since $\mathsf{isSpur}(q, \mathsf{D}(\mathcal{K}), \tau) = \mathsf{f}$, we have $\tau(v') \approx \tau(t') \in J$. Since $\approx$ is a congruence relation, we have $R(\tau(v'), \tau(t)) \in J$. By Lemma 4, for each term $w' \in \mathsf{dom}(I)$ such that $\delta(w') = \tau(v')$, we have $R(w', f_{R,A}(w')) \in I$. By property M1 of Lemma 18, we have $R(\pi(v'), f_{R,A}(\pi(v'))) \in I$, and by Property M2 of Lemma 18, we have $\pi(t') \approx \pi(v') \in I$. Therefore, we have $R(\pi(t'), f_{R,A}(\pi(v'))) \in I$. □

Proof of Theorem 11

Theorem 11. *Deciding whether $\mathcal{K} \models \pi(q)$ holds can be implemented in nondeterministic polynomial time w.r.t. the size of $\mathcal{K}$ and q, and in polynomial time w.r.t. the size of $\mathcal{A}$.*

Proof. First, we argue that we can compute relation $\sim$ in polynomial time. For each term u, we can decide whether $u \in \mathsf{aux}_{\mathsf{D}(\mathcal{K})}$ by checking whether, for each term u', we have that $\mathsf{D}(\mathcal{K}) \models u \approx u'$ implies $u' \notin N_I$. Since the number of variables occurring in each clause in $\mathsf{D}(\mathcal{K})$ is bounded by a constant, this check can be performed in polynomial time. Thus, we can evaluate in polynomial time the precondition of the (fork) rule. In addition, the size of relation $\sim$ is bounded by $|N_T(q)|^2$, the rules used to compute it are monotonic, and each inference can be applied in polynomial time, so we can compute $\sim$ in polynomial time.

Second, we show that we can decide whether q is aux-cyclic w.r.t. τ in polynomial time. Since $\sim$ can be computed in polynomial time and the size of G_{aux} is polynomially bounded by the number of terms occurring in q, we can compute G_{aux} in polynomial time. Also, we can check whether G_{aux} is a acyclic by searching for a topological ordering of its vertexes in linear time (Cormen et al. 2009).

For the NP upper bound, according to Theorem 10 checking whether $\mathcal{K} \models \pi(q)$ amounts to guessing a candidate answer τ for q in the minimal Herbrand model of $\mathsf{D}(\mathcal{K})$ such that $\tau|_{\vec{x}} = \pi$ and to checking that $\mathsf{isSpur}(q, \mathsf{D}(\mathcal{K}), \tau) = \mathsf{f}$. Since each clause in $\mathsf{D}(\mathcal{K})$ has a bounded number of variables, the minimal Herbrand model of $\mathsf{D}(\mathcal{K})$ can be computed in polynomial time. By the first two observations, we conclude that the whole process can be carried out in nondeterministic polynomial time in the combined size of $\mathsf{D}(\mathcal{K})$ and q.

For the PTIME upper bound, consider a fixed $\mathcal{ELHO}^r_\bot$ TBox $\mathcal{T}$ and a fixed conjunctive query q. For an arbitrary ABox $\mathcal{A}$, we can enumerate in polynomial time all possible answers to q in the minimal Herbrand model of $\mathsf{D}(\mathcal{T}) \cup \mathcal{A}$. Also, we can filter out those answers that are spurious in polynomial time. Finally, we just check whether π occurs in the remaining (certain) answers. □