

ONTOLOGY-BASED QUERY ANSWERING OVER DATALOG-EXPRESSIBLE RULE SETS IS UNDECIDABLE

DAVID CARRAL¹, LUCAS LARROQUE², AND MICHAËL THOMAZO²

¹LIRMM, INRIA, UNIVERSITY OF MONTPELLIER, CNRS

²INRIA, DI ENS, ENS, CNRS, PSL UNIVERSITY

MAY 29, 2025

PRELIMINARIES AND MOTIVATION

OBQE FOR EXISTENTIAL RULES

Definition: OBQE Problem

Given an ontology $\langle \mathcal{R}, \mathcal{F} \rangle$ consisting of an existential rule set \mathcal{R} and a fact set \mathcal{F} , and a fact φ ; check if $\langle \mathcal{R}, \mathcal{F} \rangle \models \varphi$ under FO semantics.

OBQE FOR EXISTENTIAL RULES

Definition: OBQE Problem

Given an ontology $\langle \mathcal{R}, \mathcal{F} \rangle$ consisting of an existential rule set \mathcal{R} and a fact set \mathcal{F} , and a fact φ ; check if $\langle \mathcal{R}, \mathcal{F} \rangle \models \varphi$ under FO semantics.

Example: Existential Rules

$$\forall x, y, z. (\text{Connected}(x, y) \wedge \text{Connected}(y, z) \rightarrow \text{Connected}(x, z))$$

$$\forall x. (\text{Human}(x) \rightarrow \exists y. \text{HasParent}(x, y) \wedge \text{Human}(y))$$

$$\forall x, y, z. (P(x, y, z) \wedge Q(x, z) \rightarrow \exists w, v. R(x, y, v, w) \wedge S(w, y) \wedge P(z, z, w))$$

OBQE FOR EXISTENTIAL RULES

Definition: OBQE Problem

Given an ontology $\langle \mathcal{R}, \mathcal{F} \rangle$ consisting of an existential rule set \mathcal{R} and a fact set \mathcal{F} , and a fact φ ; check if $\langle \mathcal{R}, \mathcal{F} \rangle \models \varphi$ under FO semantics.

Example: Existential Rules

$$\forall x, y, z. (\text{Connected}(x, y) \wedge \text{Connected}(y, z) \rightarrow \text{Connected}(x, z))$$
$$\forall x. (\text{Human}(x) \rightarrow \exists y. \text{HasParent}(x, y) \wedge \text{Human}(y))$$
$$\forall x, y, z. (P(x, y, z) \wedge Q(x, z) \rightarrow \exists w, v. R(x, y, v, w) \wedge S(w, y) \wedge P(z, z, w))$$

We ignore universal quantifiers when writing rules.

OBQE FOR EXISTENTIAL RULES

Definition: OBQE Problem

Given an ontology $\langle \mathcal{R}, \mathcal{F} \rangle$ consisting of an existential rule set \mathcal{R} and a fact set \mathcal{F} , and a fact φ ; check if $\langle \mathcal{R}, \mathcal{F} \rangle \models \varphi$ under FO semantics.

Example: Existential Rules

$$\forall x, y, z. (\text{Connected}(x, y) \wedge \text{Connected}(y, z) \rightarrow \text{Connected}(x, z))$$
$$\forall x. (\text{Human}(x) \rightarrow \exists y. \text{HasParent}(x, y) \wedge \text{Human}(y))$$
$$\forall x, y, z. (P(x, y, z) \wedge Q(x, z) \rightarrow \exists w, v. R(x, y, v, w) \wedge S(w, y) \wedge P(z, z, w))$$

We ignore universal quantifiers when writing rules.

Example: Facts

Connected(paris, montpellier)

P(c, d, e)

OBQE FOR EXISTENTIAL RULES

Definition: OBQE Problem

Given an ontology $\langle \mathcal{R}, \mathcal{F} \rangle$ consisting of an existential rule set \mathcal{R} and a fact set \mathcal{F} , and a fact φ ; check if $\langle \mathcal{R}, \mathcal{F} \rangle \models \varphi$ under FO semantics.

Example: Existential Rules

$$\forall x, y, z. (\text{Connected}(x, y) \wedge \text{Connected}(y, z) \rightarrow \text{Connected}(x, z))$$
$$\forall x. (\text{Human}(x) \rightarrow \exists y. \text{HasParent}(x, y) \wedge \text{Human}(y))$$
$$\forall x, y, z. (P(x, y, z) \wedge Q(x, z) \rightarrow \exists w, v. R(x, y, v, w) \wedge S(w, y) \wedge P(z, z, w))$$

We ignore universal quantifiers when writing rules.

Example: Facts

Connected(paris, montpellier)

P(c, d, e)

Remark

We consider the problem of entailment and not of model-checking!

ADDRESSING OBQE VIA UBCQ-REWRITINGS

Theorem

The problem of checking if an ontology entails a fact is undecidable.

ADDRESSING OBQE VIA UBCQ-REWRITINGS

Theorem

The problem of checking if an ontology entails a fact is undecidable.

However, in some cases, we can solve OBQE via rewriting into UBCQs:

Definition: UBCQ-Rewritings

- An *(existential) rule query* is a pair $\langle \mathcal{R}, \varphi \rangle$ consisting of an existential rule set \mathcal{R} and a fact φ .
- A *UBCQ-rewriting* for a rule query $\langle \mathcal{R}, \varphi \rangle$ is a UBCQ q such that $\langle \mathcal{R}, \mathcal{F} \rangle \models \varphi \iff \mathcal{F} \models q$ for every fact set \mathcal{F} .

ADDRESSING OBQE VIA UBCQ-REWRITINGS

Theorem

The problem of checking if an ontology entails a fact is undecidable.

However, in some cases, we can solve OBQE via rewriting into UBCQs:

Definition: UBCQ-Rewritings

- An *(existential) rule query* is a pair $\langle \mathcal{R}, \varphi \rangle$ consisting of an existential rule set \mathcal{R} and a fact φ .
- A *UBCQ-rewriting* for a rule query $\langle \mathcal{R}, \varphi \rangle$ is a UBCQ q such that $\langle \mathcal{R}, \mathcal{F} \rangle \models \varphi \iff \mathcal{F} \models q$ for every fact set \mathcal{F} .

Given an ontology $\langle \mathcal{R}, \mathcal{F} \rangle$ and a fact φ , we can:

1. Attempt to compute a UBCQ-rewriting q of the rule query $\langle \mathcal{R}, \varphi \rangle$.
2. Check if $\mathcal{F} \models q$.

Example: A UBCQ-Expressible Rule Query

Consider the rule set \mathcal{R} :

$$HasPart(x, y) \wedge Wheel(y) \rightarrow Vehicle(x) \quad (1)$$

$$Bicycle(x) \rightarrow \exists y. HasPart(x, y) \wedge Wheel(y) \quad (2)$$

Example: A UBCQ-Expressible Rule Query

Consider the rule set \mathcal{R} :

$$HasPart(x, y) \wedge Wheel(y) \rightarrow Vehicle(x) \quad (1)$$

$$Bicycle(x) \rightarrow \exists y. HasPart(x, y) \wedge Wheel(y) \quad (2)$$

The rule query $\langle \mathcal{R}, Vehicle(c) \rangle$ admits the following UBCQ-rewriting:

$$q = Vehicle(c) \vee (\exists y. HasPart(c, y) \wedge Wheel(y)) \vee Bicycle(c)$$

REWRITING INTO UBCQs

Example: A UBCQ-Expressible Rule Query

Consider the rule set \mathcal{R} :

$$HasPart(x, y) \wedge Wheel(y) \rightarrow Vehicle(x) \quad (1)$$

$$Bicycle(x) \rightarrow \exists y. HasPart(x, y) \wedge Wheel(y) \quad (2)$$

The rule query $\langle \mathcal{R}, Vehicle(c) \rangle$ admits the following UBCQ-rewriting:

$$q = Vehicle(c) \vee (\exists y. HasPart(c, y) \wedge Wheel(y)) \vee Bicycle(c)$$

That is, for every set \mathcal{F} of facts, we have that:

$$\langle \mathcal{R}, \mathcal{F} \rangle \models Vehicle(c) \iff \mathcal{F} \models q$$

A UNIFORM PROCEDURE FOR UBCQ-REWRITINGS

Theorem

We cannot decide if a rule query is UBCQ-expressible.

A UNIFORM PROCEDURE FOR UBCQ-REWRITINGS

Theorem

We cannot decide if a rule query is UBCQ-expressible.

Theorem (see [KLMT15] for more info)

There is a procedure that, on input q a rule query:

- Outputs a UBCQ-rewriting of q .
- Terminates if q is UBCQ-expressible.

A UNIFORM PROCEDURE FOR UBCQ-REWRITINGS

Theorem

We cannot decide if a rule query is UBCQ-expressible.

Theorem (see [KLMT15] for more info)

There is a procedure that, on input q a rule query:

- Outputs a UBCQ-rewriting of q .
- Terminates if q is UBCQ-expressible.

Remark

- In practice: Reuse the same procedure to rewrite every UBCQ-expressible fragment: DL-Lite, linear and sticky rules,...
- In theory: Use this procedure as a tool to show if a rule query is UBCQ-expressible.

THE LIMITS OF THE UBCQ-EXPRESSIBLE FRAGMENT

Example: A Datalog-Expressible Rule Query

Consider the rule set \mathcal{R} :

$$\text{Colleague}(x, y) \wedge \text{MainAff}(x, z) \wedge \text{MainAff}(y, w) \rightarrow z \approx w \quad (3)$$

$$\text{Colleague}(x, y) \rightarrow \text{Colleague}(y, x) \quad (4)$$

$$\text{Colleague}(x, y) \wedge \text{Colleague}(y, z) \rightarrow \text{Colleague}(x, z) \quad (5)$$

$$\text{Colleague}(x, y) \rightarrow \text{Academic}(x) \quad (6)$$

$$\text{Colleague}(x, y) \rightarrow \text{Academic}(y) \quad (7)$$

$$\text{Academic}(x) \rightarrow \exists y. \text{MainAff}(x, y) \quad (8)$$

THE LIMITS OF THE UBCQ-EXPRESSIBLE FRAGMENT

Example: A Datalog-Expressible Rule Query

Consider the rule set \mathcal{R} :

$$\text{Colleague}(x, y) \wedge \text{MainAff}(x, z) \wedge \text{MainAff}(y, w) \rightarrow z \approx w \quad (3)$$

$$\text{Colleague}(x, y) \rightarrow \text{Colleague}(y, x) \quad (4)$$

$$\text{Colleague}(x, y) \wedge \text{Colleague}(y, z) \rightarrow \text{Colleague}(x, z) \quad (5)$$

$$\text{Colleague}(x, y) \rightarrow \text{Academic}(x) \quad (6)$$

$$\text{Colleague}(x, y) \rightarrow \text{Academic}(y) \quad (7)$$

$$\text{Academic}(x) \rightarrow \exists y. \text{MainAff}(x, y) \quad (8)$$

The rule query $\langle \mathcal{R}, \text{MainAff}(\text{alice}, \text{inria}) \rangle$ is not UBCQ-expressible.

THE LIMITS OF THE UBCQ-EXPRESSIBLE FRAGMENT

Example: A Datalog-Expressible Rule Query

Consider the rule set \mathcal{R} :

$$\text{Colleague}(x, y) \wedge \text{MainAff}(x, z) \wedge \text{MainAff}(y, w) \rightarrow z \approx w \quad (3)$$

$$\text{Colleague}(x, y) \rightarrow \text{Colleague}(y, x) \quad (4)$$

$$\text{Colleague}(x, y) \wedge \text{Colleague}(y, z) \rightarrow \text{Colleague}(x, z) \quad (5)$$

$$\text{Colleague}(x, y) \rightarrow \text{Academic}(x) \quad (6)$$

$$\text{Colleague}(x, y) \rightarrow \text{Academic}(y) \quad (7)$$

$$\text{Academic}(x) \rightarrow \exists y. \text{MainAff}(x, y) \quad (8)$$

The rule query $\langle \mathcal{R}, \text{MainAff}(\text{alice}, \text{inria}) \rangle$ is not UBCQ-expressible.
However, we obtain a datalog-rewriting by replacing (8) in \mathcal{R} with

$$\text{Colleague}(x, y) \wedge \text{MainAff}(x, z) \rightarrow \text{MainAff}(y, z)$$

MANY PROCEDURES TO REWRITE INTO DATALOG

Source Language	Target Language	Implemented	Reference
<i>SHIQ</i>	Disj. Datalog	Yes	[HMS07]
<i>SHIQ_{b_s}</i>	Disj. Datalog	No	[RKH12]
Horn- <i>ALCHOIQ</i>	Datalog	Yes	[CDK18]
Horn- <i>SRIQ</i>	Datalog	Yes	[CGK19]
Bounded Depth Rules	Datalog	No	[Mar12]
Frontier Guarded Rules	Datalog	No	[BBtC13]
Nearly Guarded Rules	Datalog	No	[GRS14]
Guarded Disj. Rules	Disj. Datalog	No	[AOS18]
Guarded Rules	Datalog	Yes	[BBG ⁺ 22]
Warded rules	Datalog	Yes	[BGPS22]
Linear	Non-Rec. Datalog	No	[GS12]
Sticky(-Join)	Non-Rec. Datalog	No	[GS12]

A UNIFORM PROCEDURE TO REWRITE INTO DATALOG?

Research Question

Can we define a uniform rewriting procedure into datalog with strong termination guarantees?

A UNIFORM PROCEDURE TO REWRITE INTO DATALOG?

Research Question

Can we define a uniform rewriting procedure into datalog with strong termination guarantees?

Theorem

There is no procedure to check if an ontology $\langle \mathcal{R}, \mathcal{F} \rangle$ entails a fact φ that is sound, complete, and terminates if $\langle \mathcal{R}, \varphi \rangle$ is datalog-expressible.

A UNIFORM PROCEDURE TO REWRITE INTO DATALOG?

Research Question

Can we define a uniform rewriting procedure into datalog with strong termination guarantees?

Theorem

There is no procedure to check if an ontology $\langle \mathcal{R}, \mathcal{F} \rangle$ entails a fact φ that is sound, complete, and terminates if $\langle \mathcal{R}, \varphi \rangle$ is datalog-expressible.

Corollary

There is a no procedure that, on input q a rule query:

- Outputs a datalog-rewriting of q .
- Terminates if q is datalog-expressible.

CONSIDERING OTHER QUERY LANGUAGES

Problems for a Query Language \mathcal{L}

1. Is there a sound and complete procedure to rewrite rule queries into \mathcal{L} that terminates when the input is \mathcal{L} -expressible?
2. Is there a procedure to check if an ontology $\langle \mathcal{R}, \mathcal{F} \rangle$ entails a fact φ that is sound, complete, and terminates if $\langle \mathcal{R}, \varphi \rangle$ is \mathcal{L} -expressible?

CONSIDERING OTHER QUERY LANGUAGES

Problems for a Query Language \mathcal{L}

1. Is there a sound and complete procedure to rewrite rule queries into \mathcal{L} that terminates when the input is \mathcal{L} -expressible?
2. Is there a procedure to check if an ontology $\langle \mathcal{R}, \mathcal{F} \rangle$ entails a fact φ that is sound, complete, and terminates if $\langle \mathcal{R}, \varphi \rangle$ is \mathcal{L} -expressible?

Results

Language \mathcal{L}	Q1: Rewritability	Q2: Decidability
Datalog	✗	✗
Linear Datalog	✗	✗
Monadic Datalog	✗	?
Unions of CRPQs	✗	?
Unions of BCQs	✓	✓

TECHNICAL RESULTS

UNDECIDABILITY OF DAT-EXPRESSIBLE RULE QUERIES

Theorem

There is no procedure to check if an ontology $\langle \mathcal{R}, \mathcal{F} \rangle$ entails a fact φ that is sound, complete, and terminates if $\langle \mathcal{R}, \varphi \rangle$ is datalog-expressible.

UNDECIDABILITY OF DAT-EXPRESSIBLE RULE QUERIES

Theorem

There is no procedure to check if an ontology $\langle \mathcal{R}, \mathcal{F} \rangle$ entails a fact φ that is sound, complete, and terminates if $\langle \mathcal{R}, \varphi \rangle$ is datalog-expressible.

Proof. We reduce machine M to a rule set \mathcal{R}_M such that:

UNDECIDABILITY OF DAT-EXPRESSIBLE RULE QUERIES

Theorem

There is no procedure to check if an ontology $\langle \mathcal{R}, \mathcal{F} \rangle$ entails a fact φ that is sound, complete, and terminates if $\langle \mathcal{R}, \varphi \rangle$ is datalog-expressible.

Proof. We reduce machine M to a rule set \mathcal{R}_M such that:

- Lemma 1. The machine M halts on ϵ if and only if the ontology $\langle \mathcal{R}_M, \emptyset \rangle$ entails the (nullary) fact *Halt*.

UNDECIDABILITY OF DAT-EXPRESSIBLE RULE QUERIES

Theorem

There is no procedure to check if an ontology $\langle \mathcal{R}, \mathcal{F} \rangle$ entails a fact φ that is sound, complete, and terminates if $\langle \mathcal{R}, \varphi \rangle$ is datalog-expressible.

Proof. We reduce machine M to a rule set \mathcal{R}_M such that:

- Lemma 1. The machine M halts on ϵ if and only if the ontology $\langle \mathcal{R}_M, \emptyset \rangle$ entails the (nullary) fact $Halt$.
- Lemma 2. The rule query $\langle \mathcal{R}_M, Halt \rangle$ is datalog-expressible.

UNDECIDABILITY OF DAT-EXPRESSIBLE RULE QUERIES

Theorem

There is no procedure to check if an ontology $\langle \mathcal{R}, \mathcal{F} \rangle$ entails a fact φ that is sound, complete, and terminates if $\langle \mathcal{R}, \varphi \rangle$ is datalog-expressible.

Proof. We reduce machine M to a rule set \mathcal{R}_M such that:

- Lemma 1. The machine M halts on ϵ if and only if the ontology $\langle \mathcal{R}_M, \emptyset \rangle$ entails the (nullary) fact $Halt$.
- Lemma 2. The rule query $\langle \mathcal{R}_M, Halt \rangle$ is datalog-expressible.
 - If M halts, then $\langle \rightarrow Halt, Halt \rangle$ is a datalog-rewriting.

UNDECIDABILITY OF DAT-EXPRESSIBLE RULE QUERIES

Theorem

There is no procedure to check if an ontology $\langle \mathcal{R}, \mathcal{F} \rangle$ entails a fact φ that is sound, complete, and terminates if $\langle \mathcal{R}, \varphi \rangle$ is datalog-expressible.

Proof. We reduce machine M to a rule set \mathcal{R}_M such that:

- Lemma 1. The machine M halts on ϵ if and only if the ontology $\langle \mathcal{R}_M, \emptyset \rangle$ entails the (nullary) fact $Halt$.
- Lemma 2. The rule query $\langle \mathcal{R}_M, Halt \rangle$ is datalog-expressible.
 - ▶ If M halts, then $\langle \rightarrow Halt, Halt \rangle$ is a datalog-rewriting.
 - ▶ Otherwise, $\langle \mathcal{R}_M^\forall, Halt \rangle$ is a datalog-rewriting.

UNDECIDABILITY OF DAT-EXPRESSIBLE RULE QUERIES

Theorem

There is no procedure to check if an ontology $\langle \mathcal{R}, \mathcal{F} \rangle$ entails a fact φ that is sound, complete, and terminates if $\langle \mathcal{R}, \varphi \rangle$ is datalog-expressible.

Proof. We reduce machine M to a rule set \mathcal{R}_M such that:

- Lemma 1. The machine M halts on ϵ if and only if the ontology $\langle \mathcal{R}_M, \emptyset \rangle$ entails the (nullary) fact $Halt$.
- Lemma 2. The rule query $\langle \mathcal{R}_M, Halt \rangle$ is datalog-expressible.
 - ▶ If M halts, then $\langle \rightarrow Halt, Halt \rangle$ is a datalog-rewriting.
 - ▶ Otherwise, $\langle \mathcal{R}_M^\forall, Halt \rangle$ is a datalog-rewriting.

After proving the above, the theorem follows by contradiction.

UNDECIDABILITY OF DAT-EXPRESSIBLE RULE QUERIES

Theorem

There is no procedure to check if an ontology $\langle \mathcal{R}, \mathcal{F} \rangle$ entails a fact φ that is sound, complete, and terminates if $\langle \mathcal{R}, \varphi \rangle$ is datalog-expressible.

Proof. We reduce machine M to a rule set \mathcal{R}_M such that:

- Lemma 1. The machine M halts on ϵ if and only if the ontology $\langle \mathcal{R}_M, \emptyset \rangle$ entails the (nullary) fact $Halt$.
- Lemma 2. The rule query $\langle \mathcal{R}_M, Halt \rangle$ is datalog-expressible.
 - ▶ If M halts, then $\langle \rightarrow Halt, Halt \rangle$ is a datalog-rewriting.
 - ▶ Otherwise, $\langle \mathcal{R}_M^\forall, Halt \rangle$ is a datalog-rewriting.

After proving the above, the theorem follows by contradiction.

Remark

The proof for linear datalog is conceptually similar.

Theorem

There is a no procedure that, on input q a rule query:

- Outputs a CRPQ-rewriting of q .
- Terminates if q is CRPQ-expressible.

UNREWRITABILITY: CRPQ-EXPRESSIBLE QUERIES

Theorem

There is no procedure that, on input q a rule query:

- Outputs a CRPQ-rewriting of q .
- Terminates if q is CRPQ-expressible.

Proof by Contradiction. If such a procedure exists, we can semi-decide if a context-free grammar (CFG) is universal:

UNREWRITABILITY: CRPQ-EXPRESSIBLE QUERIES

Theorem

There is no procedure that, on input q a rule query:

- Outputs a CRPQ-rewriting of q .
- Terminates if q is CRPQ-expressible.

Proof by Contradiction. If such a procedure exists, we can semi-decide if a context-free grammar (CFG) is universal:

- Assume that such a rewriting procedure does exist.

UNREWRITABILITY: CRPQ-EXPRESSIBLE QUERIES

Theorem

There is no procedure that, on input q a rule query:

- Outputs a CRPQ-rewriting of q .
- Terminates if q is CRPQ-expressible.

Proof by Contradiction. If such a procedure exists, we can semi-decide if a context-free grammar (CFG) is universal:

- Assume that such a rewriting procedure does exist.
- Encode a CFG G as a datalog rule query $\langle \mathcal{R}_G, \text{Goal} \rangle$.

UNREWRITABILITY: CRPQ-EXPRESSIBLE QUERIES

Theorem

There is no procedure that, on input q a rule query:

- Outputs a CRPQ-rewriting of q .
- Terminates if q is CRPQ-expressible.

Proof by Contradiction. If such a procedure exists, we can semi-decide if a context-free grammar (CFG) is universal:

- Assume that such a rewriting procedure does exist.
- Encode a CFG G as a datalog rule query $\langle \mathcal{R}_G, \text{Goal} \rangle$.
- Apply the rewriting procedure to $\langle \mathcal{R}_G, \text{Goal} \rangle$. If G is universal, it halts and outputs an equivalent CRPQ q .

UNREWRITABILITY: CRPQ-EXPRESSIBLE QUERIES

Theorem

There is a no procedure that, on input q a rule query:

- Outputs a CRPQ-rewriting of q .
- Terminates if q is CRPQ-expressible.

Proof by Contradiction. If such a procedure exists, we can semi-decide if a context-free grammar (CFG) is universal:

- Assume that such a rewriting procedure does exist.
- Encode a CFG G as a datalog rule query $\langle \mathcal{R}_G, \text{Goal} \rangle$.
- Apply the rewriting procedure to $\langle \mathcal{R}_G, \text{Goal} \rangle$. If G is universal, it halts and outputs an equivalent CRPQ q .
- Verify that q contains all words with a containment check. The grammar G is universal if and only if this is the case.

UNREWRITABILITY: CRPQ-EXPRESSIBLE QUERIES

Theorem

There is no procedure that, on input q a rule query:

- Outputs a CRPQ-rewriting of q .
- Terminates if q is CRPQ-expressible.

Proof by Contradiction. If such a procedure exists, we can semi-decide if a context-free grammar (CFG) is universal:

- Assume that such a rewriting procedure does exist.
- Encode a CFG G as a datalog rule query $\langle \mathcal{R}_G, \text{Goal} \rangle$.
- Apply the rewriting procedure to $\langle \mathcal{R}_G, \text{Goal} \rangle$. If G is universal, it halts and outputs an equivalent CRPQ q .
- Verify that q contains all words with a containment check. The grammar G is universal if and only if this is the case.

Remark

The proof for monadic datalog is almost the same.

FUTURE WORK AND CONCLUSIONS

FUTURE WORK AND CONCLUSIONS

Problems for a Query Language \mathcal{L}

1. Is there a sound and complete procedure to rewrite rule queries into \mathcal{L} that terminates when the input is \mathcal{L} -expressible?
2. Is there a procedure to check if an ontology $\langle \mathcal{R}, \mathcal{F} \rangle$ entails a fact φ that is sound, complete, and terminates if $\langle \mathcal{R}, \varphi \rangle$ is \mathcal{L} -expressible?

Results




Language \mathcal{L}	Q1: Rewritability	Q2: Decidability
Datalog	✗	✗
Linear Datalog	✗	✗
Monadic Datalog	✗	?
Unions of CRPQs	✗	?
Unions of BCQs	✓	✓

Thanks for your attention!

REFERENCES I

-  SHQIPONJA AHMETAJ, MAGDALENA ORTIZ, AND MANTAS SIMKUS, *REWRITING GUARDED EXISTENTIAL RULES INTO SMALL DATALOG PROGRAMS*, 21ST INTERNATIONAL CONFERENCE ON DATABASE THEORY, ICDT 2018, MARCH 26-29, 2018, VIENNA, AUSTRIA (BENNY KIMELFELD AND Yael AMSTERDAMER, EDS.), LIPICS, VOL. 98, SCHLOSS DAGSTUHL - LEIBNIZ-ZENTRUM FÜR INFORMATIK, 2018, PP. 4:1–4:24.
-  MICHAEL BENEDIKT, MAXIME BURON, STEFANO GERMANO, KEVIN KAPPELMANN, AND BORIS MOTIK, *REWRITING THE INFINITE CHASE*, PROC. VLDB ENDOW. **15** (2022), NO. 11, 3045–3057.
-  VINCE BÁRÁNY, MICHAEL BENEDIKT, AND BALDER TEN CATE, *REWRITING GUARDED NEGATION QUERIES*, MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE 2013 - 38TH INTERNATIONAL SYMPOSIUM, MFCS 2013, KLOSTERNEUBURG, AUSTRIA, AUGUST 26-30, 2013. PROCEEDINGS (KRISHNENDU CHATTERJEE AND JIRÍ SGALL, EDS.), LECTURE NOTES IN COMPUTER SCIENCE, VOL. 8087, SPRINGER, 2013, PP. 98–110.

REFERENCES II

-  GERALD BERGER, GEORG GOTTLÖB, ANDREAS PIERIS, AND EMANUEL SALLINGER, *THE SPACE-EFFICIENT CORE OF VADALOG*, ACM TRANS. DATABASE SYST. **47** (2022), NO. 1, 1:1–1:46.
-  DAVID CARRAL, IRINA DRAGOSTE, AND MARKUS KRÖTZSCH, *THE COMBINED APPROACH TO QUERY ANSWERING IN HORN-ALCHOIQ*, PRINCIPLES OF KNOWLEDGE REPRESENTATION AND REASONING: PROCEEDINGS OF THE SIXTEENTH INTERNATIONAL CONFERENCE, KR 2018, TEMPE, ARIZONA, 30 OCTOBER - 2 NOVEMBER 2018 (MICHAEL THIELSCHER, FRANCESCA TONI, AND FRANK WOLTER, EDS.), AAAI PRESS, 2018, PP. 339–348.
-  DAVID CARRAL, LARRY GONZÁLEZ, AND PATRICK KOOPMANN, *FROM HORN-SRIQ TO DATALOG: A DATA-INDEPENDENT TRANSFORMATION THAT PRESERVES ASSERTION ENTAILMENT*, THE THIRTY-THIRD AAAI CONFERENCE ON ARTIFICIAL INTELLIGENCE, AAAI 2019, THE THIRTY-FIRST INNOVATIVE APPLICATIONS OF ARTIFICIAL INTELLIGENCE CONFERENCE, IAAI 2019, THE NINTH AAAI SYMPOSIUM ON EDUCATIONAL ADVANCES IN

REFERENCES III

ARTIFICIAL INTELLIGENCE, EAAI 2019, HONOLULU, HAWAII, USA, JANUARY 27 - FEBRUARY 1, 2019, AAAI PRESS, 2019, PP. 2736–2743.







GEORG GOTTLOB, SEBASTIAN RUDOLPH, AND MANTAS SIMKUS, *EXPRESSIVENESS OF GUARDED EXISTENTIAL RULE LANGUAGES*, PROCEEDINGS OF THE 33RD ACM SIGMOD-SIGACT-SIGART SYMPOSIUM ON PRINCIPLES OF DATABASE SYSTEMS, PODS'14, SNOWBIRD, UT, USA, JUNE 22-27, 2014 (RICHARD HULL AND MARTIN GROHE, EDS.), ACM, 2014, PP. 27–38.



GEORG GOTTLOB AND THOMAS SCHWENTICK, *REWRITING ONTOLOGICAL QUERIES INTO SMALL NONRECURSIVE DATALOG PROGRAMS*, PRINCIPLES OF KNOWLEDGE REPRESENTATION AND REASONING: PROCEEDINGS OF THE THIRTEENTH INTERNATIONAL CONFERENCE, KR 2012, ROME, ITALY, JUNE 10-14, 2012 (GERHARD BREWKA, THOMAS EITER, AND SHEILA A. MCILRAITH, EDS.), AAAI PRESS, 2012.

REFERENCES IV

-  ULLRICH HUSTADT, BORIS MOTIK, AND ULRIKE SATTLER, *REASONING IN DESCRIPTION LOGICS BY A REDUCTION TO DISJUNCTIVE DATALOG*, J. AUTOM. REASON. **39** (2007), NO. 3, 351–384.
-  MÉLANIE KÖNIG, MICHEL LECLÈRE, MARIE-LAURE MUGNIER, AND MICHAËL THOMAZO, *SOUND, COMPLETE AND MINIMAL UCQ-REWRITING FOR EXISTENTIAL RULES*, SEMANTIC WEB **6** (2015), NO. 5, 451–475.
-  BRUNO MARNETTE, *RESOLUTION AND DATALOG REWRITING UNDER VALUE INVENTION AND EQUALITY CONSTRAINTS*, CORR **ABS/1212.0254** (2012).
-  SEBASTIAN RUDOLPH, MARKUS KRÖTZSCH, AND PASCAL HITZLER, *TYPE-ELIMINATION-BASED REASONING FOR THE DESCRIPTION LOGIC SHIQBS USING DECISION DIAGRAMS AND DISJUNCTIVE DATALOG*, LOG. METHODS COMPUT. SCI. **8** (2012), NO. 1.