Ontology-Based Query Answering over Datalog-Expressible Rule Sets is Undecidable

David Carral[1], Lucas Larroque[2], and Michaël Thomazo[2]

[1]LIRMM, Inria, University of Montpellier, CNRS
[2]Inria, DI ENS, ENS, CNRS, PSL University

November 6, 2024

Preliminaries and Motivation

Definition: The OBQE Problem (in our Context)

Given a set $\mathcal{R}$ of existential rules, a set $\mathcal{F}$ of facts, and a fact φ; decide if $\langle \mathcal{R}, \mathcal{F} \rangle \models \varphi$ under standard first-order semantics.

A pair such as $\langle \mathcal{R}, \mathcal{F} \rangle$ above is a *knowledge base* (KB).

Example: (Existential) Rules

$$\forall x, y, z. \big(Connected(x, y) \wedge Connected(y, z) \rightarrow Connected(x, z) \big)$$

$$\forall x. \big(Human(x) \rightarrow \exists y. HasParent(x, y) \wedge Human(y) \big)$$

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

We ignore universal quantifiers when writing rules.

Example: Facts

$$Connected(paris, montpellier) \qquad P(c, d, e)$$

Theorem

We cannot decide if a KB $\langle \mathcal{R}, \mathcal{F} \rangle$ entails a fact φ.

Definition: UBCQ-Rewritings

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

We write "UBCQ" instead of "union of Boolean conjunctive queries".

Definition: Solving OBQE via Rewritings

Consider an input to the OBQE problem consisting of a KB $\langle \mathcal{R}, \mathcal{F} \rangle$ and a fact φ. In some cases, we solve this instance by:

- Computing a UBCQ-rewriting γ of the rule query $\langle \mathcal{R}, \varphi \rangle$.
- Checking if $\mathcal{F} \models \gamma$.

The decidability of reasoning for many well-known KR languages (DL-Lite, linear/sticky rules, ...) relies on UBCQ-rewritability.

Definition: UBCQ-Expressibility/Rewritability

- A rule query is *UBCQ-expressible* if it admits a *UBCQ-rewriting*.
- A class $\mathcal{C}$ of rule queries is *UBCQ-rewritable* if there is a procedure that, on input q a rule query:
  - Outputs an UBCQ-rewriting of q.
  - Terminates if $q \in \mathcal{C}$.
  - (May not terminate if $q \notin \mathcal{C}$.)

Remarks

- UBCQ-expressibility is a property of a rule query.
- UBCQ-rewritability is a property of a class of rule queries.

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.

Corollary

The class of all UBCQ-expressible rule queries is UBCQ-rewritable.

Remark

- We can reuse the very same procedure to rewrite every UBCQ-expressible fragment: DL-Lite, linear and sticky rules,...
- We can use this procedure to prove if a rule query is UBCQ-expressible.

Datalog-Expressibility and Rewritability

Definition: Datalog-Expressibility

A *datalog-rewriting* for a rule query $\langle \mathcal{R}, \varphi \rangle$ is a datalog query $\langle \mathcal{R}', \varphi \rangle$ such that, for every fact set $\mathcal{F}$,

$$\langle \mathcal{R}, \mathcal{F} \rangle \models \varphi \iff \langle \mathcal{R}', \mathcal{F} \rangle \models \varphi$$

A rule query is *datalog-expressible* if it admits a datalog-rewriting.

A *datalog query* is a rule query with a datalog rule set.

Definition: Datalog-Rewritability

A class $\mathcal{C}$ of rule queries is *datalog-rewritable* if there is a procedure that, on input q a rule query:

- Outputs an datalog-rewriting of q.
- Terminates if $q \in \mathcal{C}$.

Many Procedures to Rewrite into Datalog

Source Language	Target Language	Implemented	Reference
$\mathcal{SHIQ}$	Disj. Datalog	Yes	[HMS07]
$\mathcal{SHIQ}b_s$	Disj. Datalog	No	[RKH12]
Horn-$\mathcal{ALCHOIQ}$	Datalog	Yes	[CDK18]
Horn-$\mathcal{SRIQ}$	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 Single Procedure to Rewrite into Datalog?

Research Question

Is the datalog-expressible fragment datalog-rewritable?

Corollary

The class of all datalog-expressible rule queries is not datalog-rewritable.

Theorem

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

Reasoning with Datalog-Expressible Rule Queries: An Undecidable Problem

Theorem

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

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

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

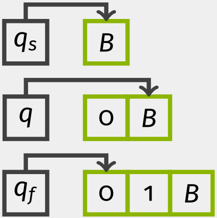
After proving the above, the theorem follows by contradiction.

EMULATING A MACHINE M WITH THE KB $\langle \mathcal{R}_M, \emptyset \rangle$

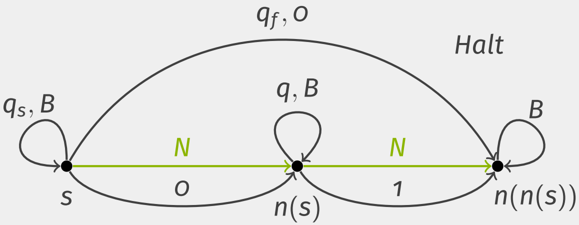
Lemma 1

The machine M halts on the empty word if and only if the KB $\langle \mathcal{R}_M, \emptyset \rangle$ entails the (nullary) fact *Halt*.

The computation of M on ϵ:



A subset of a universal model of $\langle \mathcal{R}_M, \emptyset \rangle$:



Lemma 2

The rule query $\langle \mathcal{R}_M, \mathit{Halt} \rangle$ is datalog-expressible.

Proof.

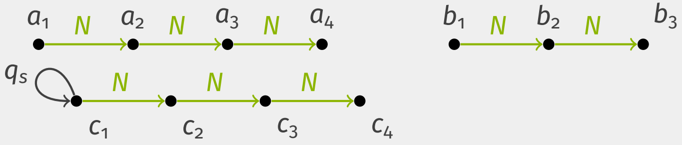
- If M halts on ϵ, then $\langle \{\rightarrow \mathit{Halt}\}, \mathit{Halt} \rangle$ is a datalog-rewriting since $\langle \mathcal{R}_M, \emptyset \rangle \models \mathit{Halt}$ by Lemma 1.
- We assume that M does not halt on ϵ.
- We can show that $\langle \mathcal{R}_M^\forall, \mathit{Halt} \rangle$ is a datalog-rewriting of $\langle \mathcal{R}_M, \mathit{Halt} \rangle$, where $\mathcal{R}_M^\forall$ is the set of datalog rules in $\mathcal{R}_M$.
- That is, $\langle \mathcal{R}_M, \mathcal{F} \rangle \models \mathit{Halt} \iff \langle \mathcal{R}_M^\forall, \mathcal{F} \rangle \models \mathit{Halt}$ for every fact set $\mathcal{F}$.
  - If $\langle \mathcal{R}_M^\forall, \mathcal{F} \rangle \models \mathit{Halt}$, then $\langle \mathcal{R}_M, \mathcal{F} \rangle \models \mathit{Halt}$ since $\mathcal{R}_M^\forall \subseteq \mathcal{R}_M$.
  - To-do: If $\langle \mathcal{R}_M, \mathcal{F} \rangle \models \mathit{Halt}$, then $\langle \mathcal{R}_M^\forall, \mathcal{F} \rangle \models \mathit{Halt}$.

The Rule Query $\langle \mathcal{R}_M, \mathit{Halt} \rangle$ is Datalog-Expressible

Lemma 3.

If M does not halt on ϵ and $\langle \mathcal{R}_M^\forall, \mathcal{F} \rangle \not\models \mathit{Halt}$, then $\langle \mathcal{R}_M, \mathcal{F} \rangle \not\models \mathit{Halt}$.

Proof. If $\langle \mathcal{R}_M^\forall, \mathcal{F} \rangle \not\models \mathit{Halt}$, then the minimal universal model of $\langle \mathcal{R}_M^\forall, \mathcal{F} \rangle$ set may look a bit like this:



We only include atoms over N and q_s in this representation.

We can extend the above into a universal model of $\langle \mathcal{R}_M, \mathcal{F} \rangle$ by:

- Extending some of the non-starting chains with one term and closing under datalog rules.
- Appending the starting chain in any universal model of $\langle \mathcal{R}_M, \emptyset \rangle$ to the starting chain above.

If M does not halt on ϵ, the resulting model does not contain Halt. $\square$

Future Work and Conclusions

Open Problems for some Query Language $\mathcal{L}$

1. Is the class of all $\mathcal{L}$-expressible queries $\mathcal{L}$-rewritable?
2. Is there a procedure to check if a KB $\langle \mathcal{R}, \mathcal{F} \rangle$ entails a fact φ that is sound, complete, and terminates if $\langle \mathcal{R}, \varphi \rangle$ is $\mathcal{L}$-expressible?

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

Conclusions

If $\mathcal{L}$ is datalog, then the answer to Q1 and Q2 above is ✗ .

Thanks for your attention!

Example: A UBCQ-Expressible Rule Query

Consider the rule set $\mathcal{R}$:

$$\mathit{MathProf}(x) \rightarrow \mathit{Prof}(x) \tag{1}$$

$$\mathit{Teaches}(x, y) \wedge \mathit{Class}(y) \rightarrow \mathit{Prof}(x) \tag{2}$$

The rule query $\langle \mathcal{R}, \mathit{Prof}(\mathit{alice}) \rangle$ admits the UBCQ-rewriting:

$$\begin{aligned} \gamma = \; & \mathit{Prof}(\mathit{alice}) \vee \mathit{MathProf}(\mathit{alice}) \vee \\ & (\exists y. \mathit{Teaches}(\mathit{alice}, y) \wedge \mathit{Class}(y)) \end{aligned}$$

That is, $\langle \mathcal{R}, \mathcal{F} \rangle \models \mathit{Prof}(\mathit{alice}) \iff \mathcal{F} \models \gamma$ for every fact set $\mathcal{F}$.

The Limits of the UBCQ-Expressible Fragment

Example: A Datalog-Expressible Rule Query

Consider the rule set $\mathcal{R}$:

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

$$\mathit{Colleague}(x, y) \rightarrow \mathit{Colleague}(y, x) \tag{4}$$

$$\mathit{Colleague}(x, y) \wedge \mathit{Colleague}(y, z) \rightarrow \mathit{Colleague}(x, z) \tag{5}$$

$$\mathit{Academic}(x) \rightarrow \exists y.\mathit{MainAff}(x, y) \tag{6}$$

The rule query $\langle \mathcal{R}, \mathit{MainAff}(\mathit{alice}, \mathit{inria}) \rangle$ is not UBCQ-expressible.
However, we obtain a datalog-rewriting by replacing (6) in $\mathcal{R}$ with

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

- 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 Gottlob, 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

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.