M2 Internship Proposal: Reasoning over Bounded First-Order Logic Ontologies

David Carral
LIRMM, Inria, University of Montpellier, CNRS, France
david.carral@inria.fr

1 Introduction

Hi! This is my best attempt to recruit some student to work with me for a couple of months starting February. Before going ahead with the technical content in this proposal (cf. Section 2), I am going to present myself and discuss some details about our potential collaboration.

Brief academic history. I did my PhD at Wright State University under the supervision of Prof. Pascal Hitzler. After my PhD, I worked as a postdoc at TU Dresden in the research group led by Prof. Markus Krötzsch. Since the beginning of 2021, I am a CRCN Inria researcher at the Boreal team at the University of Montpellier.

My research topics. Broadly speaking, I am interested in the study of logical languages (e.g., first-order logic, Description Logics, existential rules, etc), their theoretical properties, and the implementation of reasoning algorithms for these languages. To know a bit more about my work, you can check out my personal page, my DBLP profile, or my Google Scholar :

- Personal page: https://www-sop.inria.fr/members/David.Carral/
- DBLP: https://dblp.org/pid/00/11425.html
- Google: https://scholar.google.com/citations?user=5bEOMysAAAAJ

What I am looking for. I am looking for a student who is intrinsically motivated to work on some research topic that I also find interesting. I propose one such topic in this document but if you are interested in working on something else, feel free to contact me about it.

What I offer. Long story short, I offer close and active supervision to help you solve some interesting research question. On top of that, I think that I can also provide funding for up to 6 months but we are going to have to double check this later.

Prerequisite knowledge. I assume that you are somewhat familiar with the syntax and semantics of first-order logic (FOL) as well as with basic notions from computational theory such as (un)decidability, Turing machines, computational complexity, etc. On top of that, it would be great if you knew a bit about logical languages such as Description Logics or existential rules but this is not a necessary requirement.

Contact me! I am writing this document in a bit of a haste and hence, it is probably not as clear as I would like it to be. To make up for this, I encourage you to contact me if you have questions; we can schedule an online meeting to discuss any details (technical or otherwise) regarding the proposal.

2 The Research Plan

2.1 Preliminaries

- For a first-order formula β and a list $\vec{x}$ of variables, we write $\beta[\vec{x}]$ to indicate that $\vec{x}$ is the set of all free variables that occur in β. Note that we often identify a list of variables with the corresponding set.
- An *(disjunctive existential) rule* is a first-order formula of the form

$$\forall \vec{x}.\big(\beta[\vec{x}] \rightarrow \bigvee\nolimits_{i=1}^{n} \exists \vec{y}_i.\eta[\vec{x}_i, \vec{y}_i]\big) \tag{1}$$

  where $\vec{x}$, $\vec{z}$, $\vec{y}_1$, …, and $\vec{y}_n$ are pairwise disjoint lists of variables; $\vec{x}_i \subseteq \vec{x}$ for every $1 \leq i \leq n$; and β, η_1, …, and η_n are non-empty conjunc-

tions of atoms. Such a rule is *deterministic* if $n = 1$; that is, if it is disjunction-free.

- In the context of this proposal, an *ontology* is set of rules. A *deterministic ontology* is a set of deterministic rules.
- A *fact* is an atomic first-order formula $P(c_1, \ldots, c_n)$ where P is an n-ary predicates and all $c_1, \ldots, c_n$ are constants.
- A *(Boolean conjunctive) query* is a first-order formula of the form $\exists \vec{y}.\beta[\vec{y}]$ with β a conjunction of atoms. That is, a query is an existentially closed conjunction of atoms.
- A *homomorphism* is a partial function h that maps variables to terms.
  - For a term t; we define $t^h = h(t)$ if t is in the domain of h, and $t^h = t$ otherwise.
  - For an atom $P(t_1, \ldots, t_n)$, let $h(P(t_1, \ldots, t_n)) = P(t_1^h, \ldots, t_n^h)$.
  - For some sets $\mathcal{A}$ and $\mathcal{B}$ of atoms; we write $h : \mathcal{A} \to \mathcal{B}$ to indicate that the domain of h is the set of all variables in $\mathcal{A}$, and that $h(\alpha) \in \mathcal{B}$ for every $\alpha \in \mathcal{A}$.

2.2 The Problem of Ontology Based Entailment (OBE)

Here's an interesting problem, which I often consider in my own research:

ONTOLOGY BASED QUERY ENTAILMENT (OBQE)

- **Input:** an ontology $\mathcal{R}$, a fact set $\mathcal{F}$, and a query υ
- **Output:** yes iff the $\mathcal{R} \cup \mathcal{F} \models \upsilon$; that is, if $\mathcal{R} \cup \mathcal{F}$ entails υ under standard first-order semantics.

Since the above is undecidable [1], intense research efforts have been aimed at finding expressive properties of ontologies that guarantee that the problem can be solved in many real-world cases.

Rewritability. A powerful approach to retrieve decidability of OBQE is based on query rewriting techniques. The idea is to rewrite the query with respect to the ontology, and then check if the resulting rewriting is evaluated directly over the input fact set.

Definition 2.1 *Consider an ontology $\mathcal{R}$ and a query υ.*

- *A* rewriting *for $\langle \mathcal{R}, \upsilon \rangle$ is a finite set $\mathcal{Q}$ of queries such that, for every fact set $\mathcal{F}$, we have that $\mathcal{R} \cup \mathcal{F} \models \upsilon$ iff $\mathcal{F} \models \gamma$ for some $\upsilon \in \mathcal{Q}$.*
- *The tuple $\langle \mathcal{R}, \upsilon \rangle$ is* rewritable *if it admits a rewriting.*
- *The ontology $\mathcal{R}$ is* rewritable *if $\langle \mathcal{R}, \gamma \rangle$ is rewritable for every query γ.*

We present some examples to clarify the above definitions.

Example 2.1 *Consider the ontology $\mathcal{R}$ and the query υ:*

$$\mathcal{R} = \{\forall x.\mathsf{Person}(x) \rightarrow \mathsf{Mortal}(x)\} \qquad \upsilon = \exists y.\mathsf{Mortal}(y)$$

Then, the query set $\mathcal{Q} = \{\exists x.\mathsf{Mortal}(x), \exists x.\mathsf{Person}(x)\}$ is a rewriting for $\langle \mathcal{R}, \upsilon \rangle$; hence, this tuple is rewritable.

Exercise 2.1 *Argue that the ontology in the previous example is rewritable.*

Example 2.2 *Consider the ontology $\mathcal{R}$ and the query υ*[1]*:*

$$\mathcal{R} = \{\forall x, y, z.\mathsf{Path}(x, y) \wedge \mathsf{Path}(y, z) \rightarrow \mathsf{Path}(x, z)\} \qquad \upsilon = \mathsf{Path}(s, t)$$

In this case, the tuple $\langle \mathcal{R}, \upsilon \rangle$ is not rewritable.

Exercise 2.2 *Show the claim at the end of the previous example.*

Consider an input ontology $\mathcal{R}$, a fact set $\mathcal{F}$, and a query υ. Then, as previously mentioned, we can solve the OBQE problem if $\mathcal{R}$ is rewritable; that is, we can effectively verify that $\mathcal{R} \cup \mathcal{F} \models \upsilon$. If this is the case, decidability is achieved by first using an (terminating) algorithm to compute a rewriting $\mathcal{Q}$ for $\langle \mathcal{R}, \upsilon \rangle$ [2, 3], and then simply checking if $\mathcal{F} \models \exists \vec{y}.\beta$ for some $\exists \vec{y}.\beta \in \mathcal{Q}$.[2] This algorithm is sound and complete by Definition 2.3.

[1] Note that a fact is a query by definition.

[2] We have that $\mathcal{F} \models \exists \vec{y}.\beta$ if and only if there is a homomorphism $h : \beta \rightarrow \mathcal{F}$; note that we identify β, which is a conjunction, with the corresponding set of facts.

Rewritability also guarantees decidability in the presence of disjunctions. However, this property is not very expressive in the non-deterministic setting; that is, there are barely any ontologies with disjunctions that are actually rewritable. For instance, see the following example were we present a singleton ontology with a stratified rule, which is not rewritable:

Example 2.3 *Consider the (extremely simple) ontology $\mathcal{R}$ and the query υ:*

$$\mathcal{R} = \{\forall x.\mathit{Edge}(x, y) \rightarrow \mathit{Source}(x) \vee \mathit{Target}(y)\}$$
$$\upsilon = \exists x, y.\mathit{Source}(x) \wedge \mathit{Edge}(x, y) \wedge \mathit{Target}(y)$$

In this case, the tuple $\langle \mathcal{R}, \upsilon \rangle$ is not rewritable. Note that rewritings are finite by Definition 2.3, and that any valid rewriting for $\langle \mathcal{R}, \upsilon \rangle$ would have to include the following infinite set of queries:

$$\{\exists x_0, \ldots, x_n.\mathit{Source}(x_0) \wedge \bigwedge\nolimits_{i=1}^{n} \mathit{Edge}(x_{i-1}, x_i) \wedge \mathit{Target}(x_n) \mid n \geq 0\}$$

Summing things up, even though rewritability does guarantee decidability in the presence of disjunctions, this property is not very useful in practice because it is not very general. Hence, let's consider an alternative definition of this property.

Boundedness. If we are only interested in deterministic ontologies, we can achieve an equivalent definition for rewritability by checking if an ontology is recursive. Intuitively, you can realise that the ontology from Example 2.2 is recursive, since the rule in the ontology may produce facts over the predicate Path that need to be considered when deriving further consequences. In contrast, the ontology from Example 2.2 is clearly not recursive; that is, this ontology is bounded. Let's give a formal definition of boundedness via the chase algorithm, which is a bottom-up materialisation procedure that produces all of the consequences for a deterministic input ontology.

Definition 2.2 *Consider the following:*

- *For an atom set $\mathcal{A}$, a deterministic rule $\rho = \beta \rightarrow \exists \vec{y}.\eta$, and a homomorphism $h : \beta \rightarrow \mathcal{A}$;*[3] *let $\mathit{Output}(\mathcal{A}, \rho, h) = h'(\eta)$ where h' is the extension of h that maps every variable $y \in \vec{y}$ to a fresh term t_y^h unique for y and h.*

[3]Here again we identify the conjunction β of atoms with the corresponding set.

- *For an atom set $\mathcal{A}$ and a deterministic rule $\rho = \beta \rightarrow \exists \vec{y}.\eta$, let $\mathsf{Output}(\mathcal{A}, \rho)$ be the set atom that includes the set $\mathsf{Output}(\mathcal{A}, \rho, h)$ for every homomorphism $h : \beta \rightarrow \mathcal{A}$.*

- *For a deterministic ontology $\mathcal{R}$ and an atom set $\mathcal{A}$, let $\mathcal{R}(\mathcal{A})$ be the atom set $\mathcal{A} \cup \bigcup_{\rho \in \mathcal{R}} \mathsf{Output}(\mathcal{A}, \rho)$.*

- *For a deterministic ontology $\mathcal{R}$ and an atom set $\mathcal{A}$; we define $\mathsf{Chase}_0(\mathcal{R}, \mathcal{A}) = \mathcal{A}$, and $\mathsf{Chase}_i(\mathcal{R}, \mathcal{A}) = \mathcal{R}(\mathsf{Chase}_{i-1}(\mathcal{R}, \mathcal{A}))$ for every $i \geq 1$. Moreover, let $\mathsf{Chase}_\infty(\mathcal{R}, \mathcal{A}) = \bigcup_{i=0}^{n} \mathsf{Chase}_i(\mathcal{R}, \mathcal{A})$.*

Consider an input ontology $\mathcal{R}$ and a fact set $\mathcal{F}$. Then, at each step of the chase (i.e., at each $\mathsf{Chase}_i(\mathcal{R}, \mathcal{F})$) we may derive more and more consequences. If we could go all the way, then we could guarantee soundness and completeness of query entailment:

Theorem 2.1 *For an ontology $\mathcal{R}$, a fact set $\mathcal{F}$, and a query υ; we have that $\mathcal{R} \cup \mathcal{F} \models \upsilon$ if and only if $\mathsf{Chase}_\infty(\mathcal{R}, \mathcal{F}) \models \upsilon$.*

Now, using the chase as our recursive algorithm for query entailment, we are finally ready to define boundedness.

Definition 2.3 *Consider a deterministic ontology $\mathcal{R}$ and a query υ.*

- *The tuple $\langle \mathcal{R}, \upsilon \rangle$ is bounded if, for every fact set $\mathcal{F}$, there is some $k \geq 1$ such that $\mathcal{R} \cup \mathcal{F} \models \upsilon$ iff $\mathsf{Chase}_k(\mathcal{R}, \mathcal{F}) \models \upsilon$.*

- *The ontology $\mathcal{R}$ is* bounded *if $\langle \mathcal{R}, \gamma \rangle$ is bounded for every query γ.*

Intuitively, the tuple $\langle \mathcal{R}, \upsilon \rangle$ above is bounded if we can check entailment against the k-th step of the chase for some fixed $k \geq 0$, which must be independent on the fact set $\mathcal{F}$. If this is the case, the chase is not a really recursive algorithm anymore since we only need to compute the chase step operation a fixed number of times.

Exercise 2.3 *Argue that the ontologies in Example 2.1 and 2.2 are and are not bounded, respectively.*

As previously mentioned, boundedness and rewritability do coincide for deterministic ontologies.

Theorem 2.2 *For a deterministic ontology $\mathcal{R}$ and a query υ, the tuple $\langle \mathcal{R}, \upsilon \rangle$ is rewritable iff $\langle \mathcal{R}, \upsilon \rangle$ is bounded. Moreover, the ontology $\mathcal{R}$ is rewritable iff $\mathcal{R}$ is bounded.*

Exercise 2.4 (A bit hard!) *Argue the "only if" direction of the first sentence in Theorem 2.2.*

Our research goal. In the presence of disjunctions, the definitions of rewritability and boundedness do not coincide anymore;[4] in fact the latter property implies the former. The converse implication does not hold; for example, the ontology in Example 2.3 is bounded but not rewritable. So, in the non-deterministic setting, boundedness is more general than rewritability and hence this property could be way more useful in practice. But (here's the catch), does boundedness guarantee decidability of the OBQE problem in the presence of disjunctions? Indeed, this is the question that we would like to solve during your internship!

3 Contact Me!

Looking back, I can see that this proposal is perhaps a bit too technical. Unfortunately, I do not have the time to write something clearer and more intuitive. Moreover, there are probably typos here and there... To make up for this, I would like to encourage you to contact me if you have any questions about this document (my email is under my name in the first page of this document) and let's just set up a online meeting.

Also, please do not feel intimidated by the task that I have proposed. If you find it interesting, please contact me even if you have no idea how to solve it. We can discuss some strategies and see if something works out :)

References

[1] C. Beeri and M. Y. Vardi. The implication problem for data dependencies. In S. Even and O. Kariv, editors, *Automata, Languages and Programming, 8th Colloquium, Israel, 1981, Proceedings*, volume 115 of *Lecture Notes in Computer Science*, pages 73–85. Springer, 1981.

[4] I would be happy to provide you with a formal definition of boundedness for disjunctive ontologies in an online meeting; I have written enough technical stuff for the day.

[2] M. König, M. Leclère, M. Mugnier, and M. Thomazo. Sound, complete and minimal ucq-rewriting for existential rules. *Semantic Web*, 6(5):451–475, 2015.

[3] M. Leclère, M. Mugnier, and G. Pérution-Kihli. Query rewriting with disjunctive existential rules and mappings. In P. Marquis, T. C. Son, and G. Kern-Isberner, editors, *Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning, KR 2023, Rhodes, Greece, September 2-8, 2023*, pages 429–439, 2023.