Ongoing Research: A Brief Review

David Carral
LIRMM, Inria, University of Montpellier, CNRS

February 12, 2021

Outline

Outline

Published or Submitted

1. Rewriting the Description Logic $\mathcal{ALCHIQ}$ to Disjunctive $\exists$-Rules
2. Materializing Knowledge Bases via Trigger Graphs
3. A Journey to the Frontiers of Query Rewritability

Outline

Published or Submitted

1. Rewriting the Description Logic $\mathcal{ALCHIQ}$ to Disjunctive $\exists$-Rules
2. Materializing Knowledge Bases via Trigger Graphs
3. A Journey to the Frontiers of Query Rewritability

Ongoing Work

4. Descriptive Complexity of Existential Rule Languages
5. Checking Chase Termination over Ontologies of Disjunctive Existential Rules: A Very Undecidable Decision Problem
6. Sufficient Notions for (Non)Termination of the Disj. Skolem Chase

Outline

Published or Submitted

1. Rewriting the Description Logic $\mathcal{ALCHIQ}$ to Disjunctive $\exists$-Rules
2. Materializing Knowledge Bases via Trigger Graphs
3. A Journey to the Frontiers of Query Rewritability

Ongoing Work

4. Descriptive Complexity of Existential Rule Languages
5. Checking Chase Termination over Ontologies of Disjunctive Existential Rules: A Very Undecidable Decision Problem
6. Sufficient Notions for (Non)Termination of the Disj. Skolem Chase

Remark

- All of the above are sorted by “chronological” order.
- Titles (4–6) will likely be changed before publication/submission.

1. Rewriting the DL $\mathcal{ALCHIQ}$ to Disjunctive Existential Rules

Rewriting $\mathcal{ALCHIQ}$ to Disjunctive $\exists$-Rules

In a Nutshell

- We present a consequence-preserving translation from $\mathcal{ALCHIQ}$ theories into existential rule sets with a terminating chase.
- Our approach produces polynomial rule sets that contain rules of bounded size (independent of the input).

Rewriting $\mathcal{ALCHIQ}$ to Disjunctive $\exists$-Rules

In a Nutshell

- We present a consequence-preserving translation from $\mathcal{ALCHIQ}$ theories into existential rule sets with a terminating chase.
- Our approach produces polynomial rule sets that contain rules of bounded size (independent of the input).

Coauthor

- Markus Krötzsch at the Technical University of Dresden

Rewriting $\mathcal{ALCHIQ}$ to Disjunctive $\exists$-Rules

In a Nutshell

- We present a consequence-preserving translation from $\mathcal{ALCHIQ}$ theories into existential rule sets with a terminating chase.
- Our approach produces polynomial rule sets that contain rules of bounded size (independent of the input).

Coauthor

- Markus Krötzsch at the Technical University of Dresden

Status

- Accepted at IJCAI 2020
- Link: `https://iccl.inf.tu-dresden.de/web/Inproceedings3244/en`

Introductory Videos

- 5 mins: https://www.youtube.com/watch?v=O67mTVKFkco&t=13s
- 17 mins: https://www.youtube.com/watch?v=Lpsw0bn7rN4&t=102s

$\mathcal{ALCHIQ}$ to Disj. ∃-Rules: Future Work

Remark

Some of our results are superseded by (3) where we show that ∃-rule sets that terminate w.r.t. the restricted chase can express every monotonic decidable query.

Remark

Some of our results are superseded by (3) where we show that ∃-rule sets that terminate w.r.t. the restricted chase can express every monotonic decidable query.

However, (I hope that) the techniques in this paper can be reused/extended to show:

Hypothesis

If a rule set has the bounded treewidth property, then it has the final model property.

$\mathcal{ALCHIQ}$ to Disj. ∃-Rules: Future Work

Remark

Some of our results are superseded by (3) where we show that ∃-rule sets that terminate w.r.t. the restricted chase can express every monotonic decidable query.

However, (I hope that) the techniques in this paper can be reused/extended to show:

Hypothesis

If a rule set has the bounded treewidth property, then it has the final model property.

If the previous hypothesis is shown, then (I hope that) we can eventually show that:

Hypothesis

All "decidable" rule sets have the final model property.

In the above, "decidable" = BTP, FES, or FUS.

2. Materializing Knowledge Bases via Trigger Graphs

In a Nutshell

- Trigger graphs are finite DAGs that are used to guide the application of rules during the computation of the chase with the goal of avoiding redundant computation.
- Trigger graphs are analogous to query plans.

Materializing KBs via Trigger Graphs

In a Nutshell

- Trigger graphs are finite DAGs that are used to guide the application of rules during the computation of the chase with the goal of avoiding redundant computation.
- Trigger graphs are analogous to query plans.

Coauthors

- Enrico Malizia at the University of Bologna
- Efthymia Tsamoura at Samsung AI Research in Cambridge
- Jacopo Urbani at the Free University of Amsterdam

In a Nutshell

- Trigger graphs are finite DAGs that are used to guide the application of rules during the computation of the chase with the goal of avoiding redundant computation.
- Trigger graphs are analogous to query plans.

Coauthors

- Enrico Malizia at the University of Bologna
- Efthymia Tsamoura at Samsung AI Research in Cambridge
- Jacopo Urbani at the Free University of Amsterdam

Status

- Accepted at VLDB 2021
- Link: `https://arxiv.org/abs/2102.02753`

Informal Definition

A trigger graph $\mathcal{G}$ for a rule set $\mathcal{R}$ is somewhat similar to the graph of rule dependencies. The main differences are:

- Trigger graphs may contain arbitrarily many nodes *labelled* with the same rule in $\mathcal{R}$. Potentially, these structures may be infinite.
- Trigger graphs are acyclic directed graphs.

Informal Definition

A trigger graph $\mathcal{G}$ for a rule set $\mathcal{R}$ is somewhat similar to the graph of rule dependencies. The main differences are:

- Trigger graphs may contain arbitrarily many nodes *labelled* with the same rule in $\mathcal{R}$. Potentially, these structures may be infinite.
- Trigger graphs are acyclic directed graphs.

Example

Consider the rule set $\mathcal{R}$ containing all of the following:

$$\rho_1 = A(x) \to B(x) \qquad \rho_2 = B(x) \to C(x) \qquad \rho_3 = A(x) \to C(x)$$

Then, $\mathcal{G} = \langle \{u, v, w\}, \{u \to v\} \rangle$ where u is labelled with rule ρ_1, and v and w are labelled with rule ρ_2. Note that ρ_3 can be safely ignored

1. Characterisation of rule sets that admit trigger graphs:

1. Characterisation of rule sets that admit trigger graphs:

Remark

Not all rule sets admit (finite) trigger graphs. For instance, neither $\{R(x,y) \rightarrow \exists z.R(y,z)\}$ nor $\{R(x,y) \wedge R(y,z) \rightarrow R(x,z)\}$ admit such graphs!

1. Characterisation of rule sets that admit trigger graphs:

Remark

Not all rule sets admit (finite) trigger graphs. For instance, neither $\{R(x,y) \rightarrow \exists z.R(y,z)\}$ nor $\{R(x,y) \wedge R(y,z) \rightarrow R(x,z)\}$ admit such graphs!

Theorem

A rule set admits a trigger graph iff it is (uniformly)-bounded.

Trigger Graphs: Theoretical contribution

1. Characterisation of rule sets that admit trigger graphs:

Remark

Not all rule sets admit (finite) trigger graphs. For instance, neither $\{R(x,y) \to \exists z.R(y,z)\}$ nor $\{R(x,y) \wedge R(y,z) \to R(x,z)\}$ admit such graphs!

Theorem

A rule set admits a trigger graph iff it is (uniformly)-bounded.

2. We define two algorithms to compute "minimal" trigger graphs for linear and Datalog rule sets (that admit trigger graphs).

Empirical Claim

Trigger graphs can be used to develop a very efficient implementation of the chase algorithm.

Empirical Claim

Trigger graphs can be used to develop a very efficient implementation of the chase algorithm.

Some (cherry-picked) results that support our claim:

	VLog	RDFOx	GLog
LUBM	170s	115s	16s
DBpedia	41s	198s	19s
Claros	431s	2373s	122s

Trigger Graphs: Practical Contribution

Empirical Claim

Trigger graphs can be used to develop a very efficient implementation of the chase algorithm.

Some (cherry-picked) results that support our claim:

	VLog	RDFOx	GLog
LUBM	170s	115s	16s
DBpedia	41s	198s	19s
Claros	431s	2373s	122s

Remarks

- The above theories only contain Datalog rules.
- When using trigger graphs, we only remove duplicates at the end.

Trigger Graphs: Future Work

Study the notion of *cyclic trigger graphs*, which (I believe) can be used to reason over rule sets with the bounded treewidth property:

Study the notion of *cyclic trigger graphs*, which (I believe) can be used to reason over rule sets with the bounded treewidth property:

Hypothesis

Given some rule set $\mathcal{R}$ with the bounded treewidth property, there is a cyclic trigger graph $\mathcal{G}$ such that:

1. $\langle \mathcal{R}, \mathcal{D} \rangle \models \varphi \iff \mathcal{G}(\mathcal{D}) \models \varphi$ for all databases $\mathcal{D}$ and all facts φ.
2. Every node in a cycle in $\mathcal{G}$ is labelled with a Datalog rule.

Note that, in the above, we can compute $\mathcal{G}(\mathcal{D})$ because of (2).

3. A Journey to the Frontiers of Query Rewritability

In a Nutshell

- Solved the FES/FUS conjecture (for rule sets where all rules with existentially quantified variables have frontier at most one...)
- Discovered a FUS rule set with rewritings of k-exponential size.

To the Frontiers of Query Rewritability

In a Nutshell

- Solved the FES/FUS conjecture (for rule sets where all rules with existentially quantified variables have frontier at most one...)
- Discovered a FUS rule set with rewritings of k-exponential size.

Coauthors

- Jerzy Marcinkowski and Piotr Ostropolski-Nalewaja at the University of Wroclaw
- Sebastian Rudolph at the Technical University of Dresden

To the Frontiers of Query Rewritability

In a Nutshell

- Solved the FES/FUS conjecture (for rule sets where all rules with existentially quantified variables have frontier at most one...)
- Discovered a FUS rule set with rewritings of k-exponential size.

Coauthors

- Jerzy Marcinkowski and Piotr Ostropolski-Nalewaja at the University of Wroclaw
- Sebastian Rudolph at the Technical University of Dresden

Status

- Submitted to LICS 2021
- ArXiv Link: `https://arxiv.org/abs/2012.11269`

Contributions: The FES/FUS Conjecture

Hypothesis: The FES/FUS Conjecture

A rule set is bounded if and only if it is both FES and FUS.

Hypothesis: The FES/FUS Conjecture

A rule set is bounded if and only if it is both FES and FUS.

Definition: Local Theories

A rule set $\mathcal{R}$ is *local* if, for some $k \geq 1$ and all databases $\mathcal{D}$,

$$\bigcup_{\mathcal{F} \subseteq \mathcal{D}, |\mathcal{F}| \leq k} Ch(\mathcal{R}, \mathcal{F}) = Ch(\mathcal{R}, \mathcal{D})$$

Contributions: The FES/FUS Conjecture

Hypothesis: The FES/FUS Conjecture

A rule set is bounded if and only if it is both FES and FUS.

Definition: Local Theories

A rule set $\mathcal{R}$ is *local* if, for some $k \geq 1$ and all databases $\mathcal{D}$,

$$\bigcup_{\mathcal{F} \subseteq \mathcal{D}, |\mathcal{F}| \leq k} Ch(\mathcal{R}, \mathcal{F}) = Ch(\mathcal{R}, \mathcal{D})$$

Theorem: Local Theories

- If a rule set $\mathcal{R}$ is local, then it is also FUS.
- If a rule set $\mathcal{R}$ is FUS, AND $\big[$every non-Datalog rule in $\mathcal{R}$ has at most one variable in its frontier OR $\mathcal{R}$ is guarded$\big]$; then $\mathcal{R}$ is local.
- The FUS/FES conjecture holds for all local theories.

Remark

Not all classes of FUS rule sets are local. For example, the singleton rule set $\{E(x,y,y',t) \wedge R(x,t') \rightarrow \exists y''.E(x,y',y'',t'')\}$ is sticky and non-local.

To tackle the FES/FUS conjecture, we developed a more general class that captures all classes of FUS rule sets:

Distancing Rule Sets

A rule set $\mathcal{R}$ is *distancing* if there is some $k_\mathcal{R} \geq 1$ such that

$$dist_{Ch(\mathcal{R},\mathcal{D})}(c,d) \leq n \implies dist_\mathcal{D}(c,d) \leq k_\mathcal{R} \cdot n$$

for all databases $\mathcal{D}$, all $c,d \in$ Consts, and all $n \geq 1$.

Theorem

All the known decidable BDD classes are bounded degree local.

In the above, “known” = linear, multi-linear, sticky, or backwards-shy

Unfortunately, we were not able to show that the FES/FUS conjecture holds for distancing theories...

Contributions: Distancing Theories and Beyond

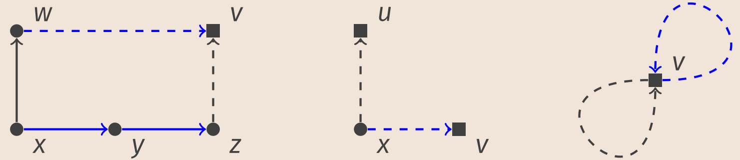
Unfortunately, we were not able to show that the FES/FUS conjecture holds for distancing theories... Nevertheless, we found out something interesting!

Contributions: Distancing Theories and Beyond

Unfortunately, we were not able to show that the FES/FUS conjecture holds for distancing theories... Nevertheless, we found out something interesting! Namely, not all FUS rule sets are distancing:

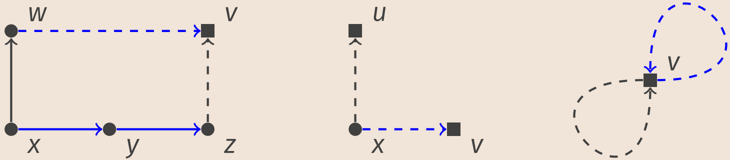
Unfortunately, we were not able to show that the FES/FUS conjecture holds for distancing theories... Nevertheless, we found out something interesting! Namely, not all FUS rule sets are distancing:

Definition: The Rule Set $\mathcal{R}$



Unfortunately, we were not able to show that the FES/FUS conjecture holds for distancing theories... Nevertheless, we found out something interesting! Namely, not all FUS rule sets are distancing:

Definition: The Rule Set $\mathcal{R}$

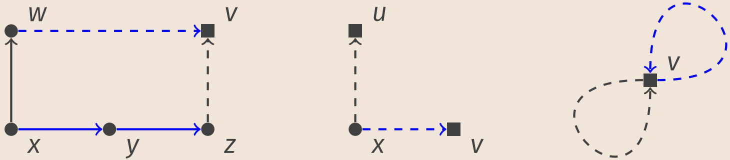


Proposition

Consider a database $\mathcal{D}$, some $c, d \in$ Consts, and some $n \geq 1$ such that $dist_{\mathcal{D}} = 2^n$. Then, it is possible that $dist_{Ch(\mathcal{R},\mathcal{D})} = 2n + 1$.

Unfortunately, we were not able to show that the FES/FUS conjecture holds for distancing theories... Nevertheless, we found out something interesting! Namely, not all FUS rule sets are distancing:

Definition: The Rule Set $\mathcal{R}$



Proposition

Consider a database $\mathcal{D}$, some $c, d \in$ Consts, and some $n \geq 1$ such that $dist_{\mathcal{D}} = 2^n$. Then, it is possible that $dist_{Ch(\mathcal{R},\mathcal{D})} = 2n + 1$.

Implication: "…there is a lot of room for new decidable/syntactic classes of BDD theories, richer than all that was considered so far."

Using the rule set from the previous slide, we can (hopefully) show:

Hypothesis

Given an ExpSpace Turing machine $\mathcal{M}$, one can compute a FUS rule set $\mathcal{R}_\mathcal{M}$ such that, for all input words $w \in \Gamma^*$, there is a BCQ γ_w such that $\langle \mathcal{R}_\mathcal{M}, \{Accept(c)\} \rangle \models \gamma_w \iff \mathcal{M}$ accepts w.

Using the rule set from the previous slide, we can (hopefully) show:

Hypothesis

Given an ExpSpace Turing machine $\mathcal{M}$, one can compute a FUS rule set $\mathcal{R}_{\mathcal{M}}$ such that, for all input words $w \in \Gamma^*$, there is a BCQ γ_w such that $\langle \mathcal{R}_{\mathcal{M}}, \{Accept(c)\} \rangle \models \gamma_w \iff \mathcal{M}$ accepts w.

Corollary

The query complexity of solving BCQ entailment over FUS rule sets is at least ExpSpace hard.

Using the rule set from the previous slide, we can (hopefully) show:

Hypothesis

Given an EXPSPACE Turing machine $\mathcal{M}$, one can compute a FUS rule set $\mathcal{R}_{\mathcal{M}}$ such that, for all input words $w \in \Gamma^*$, there is a BCQ γ_w such that $\langle \mathcal{R}_{\mathcal{M}}, \{Accept(c)\} \rangle \models \gamma_w \iff \mathcal{M}$ accepts w.

Corollary

The query complexity of solving BCQ entailment over FUS rule sets is at least EXPSPACE hard.

In fact, (I think that) we can show the above hypothesis for any kind of Turing machine...

4. Descriptive Complexity of Existential Rule Languages

In a Nutshell

- We describe the queries that can be expressed with disjunctive Datalog, weakly guarded rules, and rule sets that terminate with respect to the restricted chase.
- Using each of these rule languages, one can express all monotonic queries that are in co-NP, in ExpTime, and decidable, respectively.

Descriptive Complexity of ∃-Rule Languages

In a Nutshell

- We describe the queries that can be expressed with disjunctive Datalog, weakly guarded rules, and rule sets that terminate with respect to the restricted chase.
- Using each of these rule languages, one can express all monotonic queries that are in co-NP, in ExpTime, and decidable, respectively.

Coauthors

- Camille Bourgaux and Michaël Thomazo at ENS Paris
- Markus Krötzsch and Sebastian Rudolph at TU Dresden

Descriptive Complexity of ∃-Rule Languages

In a Nutshell

- We describe the queries that can be expressed with disjunctive Datalog, weakly guarded rules, and rule sets that terminate with respect to the restricted chase.
- Using each of these rule languages, one can express all monotonic queries that are in co-NP, in ExpTime, and decidable, respectively.

Coauthors

- Camille Bourgaux and Michaël Thomazo at ENS Paris
- Markus Krötzsch and Sebastian Rudolph at TU Dresden

Status

- Planning a submission to KR 2021
- Draft still not available

Definition

Consider some (Boolean) query γ defined over the set of all databases.

- The query γ is *closed under homomorphisms* if

$$\left[\gamma(\mathcal{D}) = \text{TRUE and there is some } h : \mathcal{D} \to \mathcal{D}'\right] \implies \gamma(\mathcal{D}') = \text{TRUE}$$

  for all databases $\mathcal{D}$ and $\mathcal{D}'$.
- The query γ is in (e.g.) NP if it can be decided with a non-deterministic poly Turing machine.

Descriptive Complexity: Queries

Definition

Consider some (Boolean) query γ defined over the set of all databases.

- The query γ is *closed under homomorphisms* if

  $$\left[\gamma(\mathcal{D}) = \text{TRUE and there is some } h : \mathcal{D} \to \mathcal{D}'\right] \implies \gamma(\mathcal{D}') = \text{TRUE}$$

  for all databases $\mathcal{D}$ and $\mathcal{D}'$.
- The query γ is in (e.g.) NP if it can be decided with a non-deterministic poly Turing machine.

Remarks

- Without loss of generality, we only consider queries that are agnostic with respect to constant names.
- We often define a query as a set of databases.

Definition

A query γ is *expressed* by a tuple $\langle \mathcal{R}, \mathit{Goal} \rangle$ with $\mathcal{R}$ a rule set and *Goal* a nullary predicate if $\langle \mathcal{R}, \mathcal{D} \rangle \models \mathit{Goal} \iff \mathcal{D} \in \gamma$ for all databases $\mathcal{D}$.

Descriptive Complexity: Expressing Queries

Definition

A query γ is *expressed* by a tuple $\langle \mathcal{R}, \mathit{Goal} \rangle$ with $\mathcal{R}$ a rule set and *Goal* a nullary predicate if $\langle \mathcal{R}, \mathcal{D} \rangle \models \mathit{Goal} \iff \mathcal{D} \in \gamma$ for all databases $\mathcal{D}$.

Example

Let γ be the set of all databases that (i) are defined over the binary predicate *Edge* and (ii) are not three-colourable. This query is *expressed* by $\langle \mathcal{R}, \mathit{N3C} \rangle$ where $\mathcal{R}$ is the rule set containing:

$$\mathit{Edge}(x, y) \to \mathit{Node}(x) \wedge \mathit{Edge}(y, x) \qquad R(x) \wedge \mathit{Edge}(x, y) \wedge G(y) \to \mathit{N3C}$$

$$\mathit{Node}(x) \to R(x) \vee G(x) \vee B(x) \qquad G(x) \wedge \mathit{Edge}(x, y) \wedge B(y) \to \mathit{N3C}$$

$$B(x) \wedge \mathit{Edge}(x, y) \wedge R(y) \to \mathit{N3C}$$

Definition

A query γ is *expressed* by a tuple $\langle\mathcal{R}, \mathit{Goal}\rangle$ with $\mathcal{R}$ a rule set and *Goal* a nullary predicate if $\langle\mathcal{R}, \mathcal{D}\rangle \models \mathit{Goal} \iff \mathcal{D} \in \gamma$ for all databases $\mathcal{D}$.

Example

Let γ be the set of all databases that (i) are defined over the binary predicate *Edge* and (ii) are not three-colourable. This query is *expressed* by $\langle\mathcal{R}, \mathit{N3C}\rangle$ where $\mathcal{R}$ is the rule set containing:

$$\mathit{Edge}(x,y) \to \mathit{Node}(x) \wedge \mathit{Edge}(y,x) \qquad R(x) \wedge \mathit{Edge}(x,y) \wedge G(y) \to \mathit{N3C}$$

$$\mathit{Node}(x) \to R(x) \vee G(x) \vee B(x) \qquad G(x) \wedge \mathit{Edge}(x,y) \wedge B(y) \to \mathit{N3C}$$

$$B(x) \wedge \mathit{Edge}(x,y) \wedge R(y) \to \mathit{N3C}$$

Hypothesis

Homomorphism-closed, decidable queries are expressible with rule sets with a terminating restricted chase. Therefore, this class of rule sets is as expressive as any other decidable FOL fragment.

5. Checking Chase Termination over Ontologies of Disjunctive Existential Rules: A Very Undecidable Decision Problem

Chase Termination over Disjunctive ∃-Rules

In a Nutshell

- We show that deciding if the oblivious/restricted/core chase variant terminate on an input ontology is an RE-complete problem.
- Furthermore, we show that deciding if the oblivious/restricted/core chase variant universally terminates for a rule set is Π^0_2-complete.

Chase Termination over Disjunctive ∃-Rules

In a Nutshell

- We show that deciding if the oblivious/restricted/core chase variant terminate on an input ontology is an RE-complete problem.
- Furthermore, we show that deciding if the oblivious/restricted/core chase variant universally terminates for a rule set is Π_2^0-complete.

Coauthor

- Markus Krötzsch at the Technical University of Dresden

Chase Termination over Disjunctive ∃-Rules

In a Nutshell

- We show that deciding if the oblivious/restricted/core chase variant terminate on an input ontology is an RE-complete problem.
- Furthermore, we show that deciding if the oblivious/restricted/core chase variant universally terminates for a rule set is Π_2^0-complete.

Coauthor

- Markus Krötzsch at the Technical University of Dresden

Status

- Hope to finish during/after the summer of 2021
- Draft still not available

Definition

A language $\mathcal{L}$ is *recursively enumerable* (RE) if it can be *recognised* by a Turing machine. That is, if there is a Turing machine (TM) $\mathcal{M}$ such that:

- For all $w \in \mathcal{L}$, the TM $\mathcal{M}$ accepts w.
- For all $w \notin \mathcal{L}$, the TM $\mathcal{M}$ rejects or does not halt on w.

A language $\mathcal{L}$ is in Π_2^0 if the complement of $\mathcal{L}$ can be recognised with an oracle TM (OTM) with an RE-oracle.

Definition

A language $\mathcal{L}$ is *recursively enumerable* (RE) if it can be *recognised* by a Turing machine. That is, if there is a Turing machine (TM) $\mathcal{M}$ such that:

- For all $w \in \mathcal{L}$, the TM $\mathcal{M}$ accepts w.
- For all $w \notin \mathcal{L}$, the TM $\mathcal{M}$ rejects or does not halt on w.

A language $\mathcal{L}$ is in Π_2^0 if the complement of $\mathcal{L}$ can be recognised with an oracle TM (OTM) with an RE-oracle.

Remark

The canonical decision problem in Π_2^0 is that of universal Turing machine halting *UH*, which is formally defined as follows:

$$\{\mathcal{M} \mid \text{there exists a word } w \in \Gamma^* \text{ such that } \mathcal{M} \text{ does not halt on } w\}$$

Chase Termination: The Decision Problems

Definition

Let $\star$ be one of the following variants of the chase algorithm: oblivious, restricted, and core.

- Let $OT^\star_{\exists\vee}$ (resp. $OT^\star_{\forall\vee}$) be the set of all ontologies for which the $\star$-chase sometimes (resp. always) terminates.
- Let $RT^\star_{\exists\vee}$ (resp. $RT^\star_{\forall\vee}$) be set of all rule sets $\mathcal{R}$ such that, for all databases $\mathcal{D}$, the $\star$-chase of the ontology $\langle\mathcal{R},\mathcal{D}\rangle$ sometimes (resp. always) terminates.
- Let $OT^\star_{\exists\wedge}$, $OT^\star_{\forall\wedge}$, $RT^\star_{\exists\wedge}$, and $RT^\star_{\forall\wedge}$ be the maximal subsets of $OT^\star_{\exists\vee}$, $OT^\star_{\forall\vee}$, $RT^\star_{\exists\vee}$, and $RT^\star_{\forall\vee}$, respectively, without rules with disjunctions.

Definition

Let $\star$ be one of the following variants of the chase algorithm: oblivious, restricted, and core.

- Let $OT^{\star}_{\exists\vee}$ (resp. $OT^{\star}_{\forall\vee}$) be the set of all ontologies for which the $\star$-chase sometimes (resp. always) terminates.
- Let $RT^{\star}_{\exists\vee}$ (resp. $RT^{\star}_{\forall\vee}$) be set of all rule sets $\mathcal{R}$ such that, for all databases $\mathcal{D}$, the $\star$-chase of the ontology $\langle \mathcal{R}, \mathcal{D} \rangle$ sometimes (resp. always) terminates.
- Let $OT^{\star}_{\exists\wedge}$, $OT^{\star}_{\forall\wedge}$, $RT^{\star}_{\exists\wedge}$, and $RT^{\star}_{\forall\wedge}$ be the maximal subsets of $OT^{\star}_{\exists\vee}$, $OT^{\star}_{\forall\vee}$, $RT^{\star}_{\exists\vee}$, and $RT^{\star}_{\forall\vee}$, respectively, without rules with disjunctions.

Definition

A rule set $\mathcal{R}$ is ($\dagger$)-*restricted* if $\mathcal{R}$ contains at most two rules with existential quantifiers and one with disjunctions, and all rules in $\mathcal{R}$ have at most 2 variables in the head.

Previous Results

1. $OT^{oblv}_{*\wedge}$, $OT^{rest}_{*\wedge}$, and $OT^{core}_{*\wedge}$ are in RE.
2. $OT^{oblv}_{*\wedge}$, $OT^{rest}_{*\wedge}$, and $RT^{oblv}_{*\wedge}$ are RE-hard even for (†)-restricted sets.
3. $OT^{core}_{*\wedge}$ is RE-hard.
4. $RT^{oblv}_{*\wedge}$ is in RE.
5. $RT^{rest}_{\exists\wedge}$ and $RT^{core}_{*\wedge}$ are Π^0_2-complete.
6. $RT^{rest}_{\forall\wedge}$ is Π^0_2-complete if we allow one denial constraint.

Previous Results

1. $OT^{oblv}_{*\wedge}$, $OT^{rest}_{*\wedge}$, and $OT^{core}_{*\wedge}$ are in RE.
2. $OT^{oblv}_{*\wedge}$, $OT^{rest}_{*\wedge}$, and $RT^{oblv}_{*\wedge}$ are RE-hard even for (†)-restricted sets.
3. $OT^{core}_{*\wedge}$ is RE-hard.
4. $RT^{oblv}_{*\wedge}$ is in RE.
5. $RT^{rest}_{\exists\wedge}$ and $RT^{core}_{*\wedge}$ are Π^0_2-complete.
6. $RT^{rest}_{\forall\wedge}$ is Π^0_2-complete if we allow one denial constraint.

New Results

1. $OT^{oblv}_{*\vee}$, $OT^{rest}_{*\vee}$, and $OT^{core}_{*\vee}$ are in RE.
2. $OT^{oblv}_{*\wedge}$, $OT^{rest}_{*\wedge}$, and $OT^{core}_{*\wedge}$ are RE-hard even for (†)-restricted sets.
3. $RT^{oblv}_{*\vee}$, $RT^{rest}_{*\vee}$, and $RT^{core}_{*\vee}$ are in Π^0_2.
4. $RT^{oblv}_{*\vee}$, $RT^{rest}_{*\wedge}$, and $RT^{core}_{*\wedge}$ are Π^0_2-hard even for (†)-restricted sets.

Chase Termination: Contribution

Previous Results

1. $OT^{oblv}_{*\wedge}$, $OT^{rest}_{*\wedge}$, and $OT^{core}_{*\wedge}$ are in RE.
2. $OT^{oblv}_{*\wedge}$, $OT^{rest}_{*\wedge}$, and $RT^{oblv}_{*\wedge}$ are RE-hard even for (†)-restricted sets.
3. $OT^{core}_{*\wedge}$ is RE-hard.
4. $RT^{oblv}_{*\wedge}$ is in RE.
5. $RT^{rest}_{\exists\wedge}$ and $RT^{core}_{*\wedge}$ are Π^0_2-complete.
6. $RT^{rest}_{\forall\wedge}$ is Π^0_2-complete if we allow one denial constraint.

New Results

1. $OT^{oblv}_{*\vee}$, $OT^{rest}_{*\vee}$, and $OT^{core}_{*\vee}$ are in RE.
2. $OT^{oblv}_{*\wedge}$, $OT^{rest}_{*\wedge}$, and $OT^{core}_{*\wedge}$ are RE-hard even for (†)-restricted sets.
3. $RT^{oblv}_{*\vee}$, $RT^{rest}_{*\vee}$, and $RT^{core}_{*\vee}$ are in Π^0_2.
4. $RT^{oblv}_{*\vee}$, $RT^{rest}_{*\wedge}$, and $RT^{core}_{*\wedge}$ are Π^0_2-hard even for (†)-restricted sets.

6. Sufficient Conditions for Termination and Nontermination of the Disjunctive Skolem Chase

In a Nutshell

- We define some sufficient conditions that guarantee universal termination and nontermination for a rule set with respect to the disjunctive Skolem chase.
- Hopefully, we will empirically demonstrate that these notions are quite general in practice.

Sufficient Conditions for the Skolem Chase

In a Nutshell

- We define some sufficient conditions that guarantee universal termination and nontermination for a rule set with respect to the disjunctive Skolem chase.
- Hopefully, we will empirically demonstrate that these notions are quite general in practice.

Coauthors

- Lukas Gerlach and Markus Krötzsch at TU Dresden

Sufficient Conditions for the Skolem Chase

In a Nutshell

- We define some sufficient conditions that guarantee universal termination and nontermination for a rule set with respect to the disjunctive Skolem chase.
- Hopefully, we will empirically demonstrate that these notions are quite general in practice.

Coauthors

- Lukas Gerlach and Markus Krötzsch at TU Dresden

Status

- Planning to finish during/after the summer of 2021
- Link to Lukas' Grosser Beleg: `https://iccl.inf.tu-dresden.de/web/Thema3509/en`

Previous Work: Empirical results for MOWLCorp

Results for Rule Sets w/o Disjunctions

#∃	#	MSA	MFA	RMSA	RMFA	RMFC	open
1–4	443	293	293	314	314	127	2
5–69	368	243	243	272	272	72	24
70–1K	409	348	348	350	350	40	19
1–1K	**1220**	**884**	**884**	**936**	**936**	**239**	**45 (3.6%)**

Previous Work: Empirical results for MOWLCorp

Results for Rule Sets w/o Disjunctions

#∃	#	MSA	MFA	RMSA	RMFA	RMFC	open
1–4	443	293	293	314	314	127	2
5–69	368	243	243	272	272	72	24
70–1K	409	348	348	350	350	40	19
1–1K	**1220**	**884**	**884**	**936**	**936**	**239**	**45 (3.6%)**

Results for Rule Sets with Disjunctions

#∃	#	MSA	MFA	RMSA	RMFA	RMFC	open
1–9	128	48	48	53	53	3	72
10–59	110	19	19	39	40	5	65
60–1K	118	23	23	30	30	20	68
1– 1K	**356**	**90**	**90**	**122**	**123**	**28**	**205 (57.6%)**

DMFA: A Novel Conditions for Termination

As RMFA, as our novel notion also makes use of "blocking" to verify that some triggers are never applied.

Example

Consider the rule set $\mathcal{R}$ that contains the following rules:

$$\rho_1 = \mathit{Pizza}(x) \rightarrow \mathit{InFridge}(x) \vee \exists y.(\mathit{DeliveryService}(y) \wedge \mathit{Delivers}(y, x))$$

$$\rho_2 = \mathit{PizzaFan}(x) \rightarrow \exists z.(\mathit{Pizza}(z) \wedge \mathit{InFridge}(z) \wedge \mathit{Owns}(x, z))$$

For every term t, the trigger $\langle \rho_1, [x/f^z(t)] \rangle$ is *blocked*. Therefore, this trigger is never applied during the computation of the disjunctive Skolem chase over any ontology of the form $\langle \mathcal{R}, \mathcal{D} \rangle$.

DMFA: A Novel Conditions for Termination

As RMFA, as our novel notion also makes use of "blocking" to verify that some triggers are never applied.

Example

Consider the rule set $\mathcal{R}$ that contains the following rules:

$$\rho_1 = \mathit{Pizza}(x) \rightarrow \mathit{InFridge}(x) \vee \exists y.(\mathit{DeliveryService}(y) \wedge \mathit{Delivers}(y, x))$$
$$\rho_2 = \mathit{PizzaFan}(x) \rightarrow \exists z.(\mathit{Pizza}(z) \wedge \mathit{InFridge}(z) \wedge \mathit{Owns}(x, z))$$

For every term t, the trigger $\langle \rho_1, [x/f^z(t)] \rangle$ is *blocked*. Therefore, this trigger is never applied during the computation of the disjunctive Skolem chase over any ontology of the form $\langle \mathcal{R}, \mathcal{D} \rangle$.

Encouraging Results!

Out of 110 rule sets in the ontology in the Oxford ontology library, only 50 are characterised as acyclic by MFA. Our novel notion (i.e., DMFA), characterises 60 of these rule sets as terminating.

Thank you for your attention!

Any questions about (1–8)?

1. Rewriting the DL $\mathcal{ALCHIQ}$ to Disjunctive $\exists$-Rules
2. Materializing Knowledge Bases via Trigger Graphs
3. A Journey to the Frontiers of Query Rewritability
4. Descriptive Complexity of Existential Rule Languages
5. Checking Chase Termination over Ontologies of Disjunctive Existential Rules: A Very Undecidable Decision Problem
6. Sufficient Conditions for (Non)Termination of the Disjunctive Skolem Chase
7. Extra Topic 1: Decidable Fragments of HyperLTL
8. Extra Topic 2: Computing Cores for the $\mathcal{ALC}$