Combining Decidability Paradigms for Existential Rules

Georg Gottlob\textsuperscript{1}  Marco Manna\textsuperscript{2}  Andreas Pieris\textsuperscript{1}

\textsuperscript{1}Department of Computer Science, University of Oxford, UK
\textsuperscript{2}Department of Mathematics, University of Calabria, Italy

ICLP 2013, Istanbul, August 26
Ontological Query Answering

\[ \langle D, O \rangle \models \text{Query} \iff D \land O \models \text{Query} \]
Ontological Query Answering: Example

\[ D = \text{person}(\text{john}) \quad O = \{ \forall P \text{ person}(P) \rightarrow \exists F \text{ father}(F,P) \}
\]

\[ \forall M \models \langle D, O \rangle = \{ \forall F \forall P \text{ father}(F,P) \rightarrow \text{person}(P) \}
\]

\[ \forall M \models \langle D, O \rangle = \{ \text{father}(z,\text{john}) \text{ person}(z) \} \]
Ontological Query Answering: Example

\[ D = \text{person}(\text{john}) \quad O = \begin{align*} &\forall P \text{ person}(P) \rightarrow \exists F \text{ father}(F, P) \\ &\forall F \forall P \text{ father}(F, P) \rightarrow \text{person}(P) \end{align*} \]

\[ \forall M \models \langle D, O \rangle = \begin{align*} &\exists x \text{ father}(x, \text{john}) \land \text{person}(x) \end{align*} \]
Ontological Query Answering: Example

\[ D = \text{person}(\text{john}) \]

\[ O = \forall P \text{ person}(P) \rightarrow \exists F \text{ father}(F,P) \]
\[ \forall F \forall P \text{ father}(F,P) \rightarrow \text{person}(P) \]

\[ \forall M \models \langle D, O \rangle = \ldots \text{father}(z, \text{john}) \text{ person}(z) \ldots \]

\[ \exists x \text{ father}(x, \text{john}) \land \text{person}(x) \]

\[ \exists x \text{ father}(\text{john}, x) \]

\[ \checkmark \]

\[ \times \]
Ontology and Query Language

\[ \forall X \varphi(X) \rightarrow \exists Y \psi(X,Y) \]

\[ \exists X \varphi(X) \]
Existential Rules: The Key Property

Existence of a universal model of $\langle D, O \rangle$

\[
\forall M \ (M \models \langle D, O \rangle \Rightarrow U \xrightarrow{\text{hom}} M)
\]
Existential Rules: The Key Property

\[ \langle D, O \rangle \models Q \iff U \models Q \]
The Chase Procedure

**Input:** Database \( D \), ontology \( O \)

**Output:** A universal model of \( \langle D, O \rangle \)

\[
D = \text{person(john)}
\]

\[
O = \text{person}(P) \rightarrow \exists F \text{ father}(F,P) \quad \text{father}(F,P) \rightarrow \text{person}(F)
\]

\[
\text{chase}(D,O) = D \cup ?
\]
The Chase Procedure

Input: Database \( D \), ontology \( O \)
Output: A universal model of \( \langle D, O \rangle \)

\[
D = \{ \text{person(john)} \}
\]

\[
O = \begin{align*}
\text{person}(P) & \rightarrow \exists F \text{ father}(F,P) \\
\text{father}(F,P) & \rightarrow \text{person}(F)
\end{align*}
\]

\[
\text{chase}(D,O) = D \cup \{ \text{father}(z_1,\text{john}) \}
\]
The Chase Procedure

**Input**: Database $D$, ontology $O$

**Output**: A universal model of $\langle D, O \rangle$

\[
D = \{ \text{person(john)} \}
\]

\[
O = \text{person}(P) \rightarrow \exists F \text{ father}(F,P) \quad \text{father}(F,P) \rightarrow \text{person}(F)
\]

\[
\text{chase}(D,O) = D \cup \{ \text{father}(z_1,\text{john}), \text{person}(z_1) \}
\]
The Chase Procedure

Input: Database $D$, ontology $O$

Output: A universal model of $\langle D, O \rangle$

$D \ = \ \text{person(john)}$

$O \ = \ \text{person}(P) \rightarrow \exists F \ \text{father}(F,P) \quad \text{father}(F,P) \rightarrow \text{person}(F)$

$\text{chase}(D,O) = D \cup \{ \text{father}(z_1,john), \text{person}(z_1), \text{father}(z_2,z_1) \}$
The Chase Procedure

Input: Database $D$, ontology $O$

Output: A universal model of $\langle D, O \rangle$

\[
D = \{ \text{person(john)} \}
\]

\[
O = \{ \text{person}(P) \rightarrow \exists F \, \text{father}(F, P), \text{father}(F, P) \rightarrow \text{person}(F) \}
\]

\[
\text{chase}(D, O) = D \cup \{ \text{father}(z_1, \text{john}), \text{person}(z_1), \text{father}(z_2, z_1), \ldots \}
\]

least fixpoint
In general is infinite

\[ D = \{ p(a,b) \} \quad p(X,Y) \rightarrow \exists Z p(Y,Z) \]

Solution = \{ p(a,b), p(b,z_1), p(z_1,z_2), p(z_2,z_3), \ldots \}

Query answering under existential rules is undecidable

implicit in [Beeri & Vardi, ICALP 1981]

\[ \text{... syntactic restrictions are needed!} \]
Weak-acyclicity

- Graph-based definition - dependency graph

\[ p(X, Y) \rightarrow \exists Z p(Y, Z) \]

\[ p[1] \xrightarrow{p[2]} \]

\[ p[1] \rightarrow \exists Z p(X, Z) \]

- Guarantees termination of the chase \(\Rightarrow\) query answering is decidable

[Fagin, Kolaitis, Miller & Popa, *Theoretical Computer Science 2005*]
Guardedness

- There exists a body-atom that contains all the body-variables

\[ \text{employee}(X), \text{supervisorOf}(X,Y), \text{manager}(Y) \rightarrow \text{manager}(X) \]

- Chase has finite treewidth \(\Rightarrow\) query answering is decidable

[Calì, Gottlob & Kifer, KR 2008]
Stickiness

- Join-variables **stick** to the inferred atoms

- Proof-theoretic procedures $\Rightarrow$ query answering is **decidable**

[Calì, Gottlob & P., Artificial Intelligence 2010]
Combining Decidability Paradigms

- **Glut-guardedness** - guard only harmful variables
  [Krötzsch & Rudolph, IJCAI 2011]

- **Weak-stickiness** - only harmful join-variables stick to the inferred atoms
  [Calì, Gottlob & P., Artificial Intelligence 2010]

- **Guardedness + stickiness** - the subject of this work
Guarded $\cup$ Sticky ($G|S$)

- $O_g$ is guarded
- $O_s$ is sticky

$O$ is a $G|S$ ontology
Theorem: Query answering under G|S is undecidable.

Proof: By reduction from query answering under general existential rules.
Source of Undecidability

Guard predicate may store non-treelike structures

\[ D = \{ p(a,b) \} \]

\[ O = \left\{ \begin{array}{l}
g(X,Y), s(X) \rightarrow s(Y) \\
p(X,Y) \rightarrow \exists z \, p(Y,Z), s(X) \\
s(X), s(Y) \rightarrow g(X,Y) \end{array} \right\} \]
Source of Undecidability

Guard predicate may store non-treelike structures

\[ D = \{ p(a, b) \} \]

\[ O = \begin{cases} 
  g(X, Y), s(X) \rightarrow s(Y) \\
  p(X, Y) \rightarrow \exists z \ p(Y, Z), s(X) \\
  s(X), s(Y) \rightarrow g(X, Y) 
\end{cases} \]

\[ \text{|nullsOf(chase}(D, O))\text{|-clique} \]

infinite
Taming the Interaction

- tame $G|S$ ontology $O$
- partition
  - $O_{g}$ is guarded
  - $O_{s}$ is sticky

- unrestricted interaction
- tamed interaction
  - do not feed the guard atom
  - may feed the non-guard atoms

  e.g., predicate-tameness
Taming the Interaction: Decidability

Encode the relevant sticky knowledge into the guarded part

unrestricted interaction → tamed interaction → unrestricted interaction

No interaction
Taming the Interaction: Decidability

unrestricted interaction

\( O_g \)

\( O_s \)

tamed interaction

encode the relevant sticky knowledge into the guarded part

\( O_g \)

\( O_s \)

unrestricted interaction

\( O_g \)

\( O_s \)

no interaction

... but what about the complexity?
The Guarded Case

Guarded Chase Forest

\[ D = \{p(a, b), s(b)\} \]

\[ O = \left\{ \begin{align*}
p(X, Y), s(Y) & \rightarrow \exists Z p(Z, X) \\
p(X, Y) & \rightarrow s(X)\end{align*} \right\} \]
The Guarded Case

Guarded Chase Forest

\[ D = \{ p(a,b), s(b) \} \]

\[ O = \{ p(X,Y), s(Y) \rightarrow \exists Z p(Z,X), p(X,Y) \rightarrow s(X) \} \]

restriction to guards and their children

\[
\begin{align*}
p(a,b) & \quad p(a,b) \\
s(b) & \quad s(b) \\
p(z_1,a) & \quad p(z_1,a) \\
s(a) & \quad s(a) \\
p(z_2,z_1) & \quad p(z_2,z_1) \\
s(z_1) & \quad s(z_1) \\
p(z_3,z_2) & \quad p(z_3,z_2) \\
s(z_2) & \quad s(z_2)
\end{align*}
\]

\[
\begin{align*}
p(a,b) & \quad p(a,b) \\
s(b) & \quad s(b) \\
p(z_1,a) & \quad p(z_1,a) \\
s(a) & \quad s(a) \\
p(z_2,z_1) & \quad p(z_2,z_1) \\
s(z_1) & \quad s(z_1) \\
p(z_3,z_2) & \quad p(z_3,z_2) \\
s(z_2) & \quad s(z_2)
\end{align*}
\]
The Guarded Case

Type of an atom

\[ \text{type}(\alpha, D, O) = \{ \beta \in \text{chase}(D, O) \mid \text{termsOf}(\beta) \subseteq \text{termsOf}(\alpha) \} \]
The Guarded Case

Type of an atom

\[
\text{type}(\alpha, D, O) = \{ \beta \in \text{chase}(D, O) \mid \text{termsOf}(\beta) \subseteq \text{termsOf}(\alpha) \}
\]
The Guarded Case

An alternating algorithm

- Guess the image of the given query - $p(z_1, c, z_2), s(z_3, z_4), p(z_1, z_3, z_5)$
- Guess the order of null generation - $z_1 \prec z_2 \quad z_1 \prec z_3 \prec z_4 \quad z_1 \prec z_3 \prec z_5$

$q(..., X, ...), t(..., Y, ...) \rightarrow s(X, Y)$

non-guarded
The Guarded Case

An alternating algorithm

- Guess the image of the given query - \( p(z_1,c,z_2), s(z_3,z_4), p(z_1,z_3,z_5) \)
- Guess the order of null generation - \( z_1 \prec z_2 \quad z_1 \prec z_3 \prec z_4 \quad z_1 \prec z_3 \prec z_5 \)
- Universally prove the relevant chase derivations

\[
\begin{align*}
|\text{type}(\alpha, D, O)| & \leq \#\text{pred} \cdot \text{maxarity}^{\text{maxarity}} \\
\Downarrow \\
\text{AEXPSPACE} & = \text{2EXPTIME (combined)} \\
\text{APSPACE} & = \text{EXPTIME (bounded arity)} \\
\text{ALOGSPACE} & = \text{PTIME (data)}
\end{align*}
\]
The Tamed Case - Difficulty I

Determine the subtree of an atom - the type is not enough

\[ p_1(b) \]
\[ p_4(b, z_1) \]
\[ p_5(b, z_1, z_2) \]
\[ p_6(z_1, z_3, b) \]
\[ p_7(b, z_1, z_3, z_2) \]
\[ p_8(b, z_1) \]
\[ p_9(z_1, z_4) \]

\[ p_3(c, b) \]

\[ \rho_1 : p_1(X) \rightarrow \exists Y p_4(X, Y) \]
\[ \rho_2 : p_4(X, Y) \rightarrow \exists Z \exists W p_5(X, Y, Z), p_6(Y, W, X) \]
\[ \rho_3 : p_5(X, Y, Z), p_8(X, Y) \rightarrow \exists W p_9(Y, W) \]
\[ \rho_4 : p_5(X, Y, Z), p_6(Y, W, X) \rightarrow p_7(X, Y, W, Z) \]
\[ \rho_5 : p_7(X, Y, Z, W), p_3(V, X) \rightarrow p_8(X, Y) \]
The Tamed Case - Difficulty I

Determine the subtree of an atom - the type is not enough

\[ p_1(b) \]
\[ p_4(b, z_1) \]
\[ p_5(b, z_1, z_2) \]
\[ p_6(z_1, z_3, b) \]
\[ p_7(b, z_1, z_3, z_2) \]
\[ p_8(b, z_1) \]
\[ p_9(z_1, z_4) \]

\[ p_3(c, b) \]

not in the type of \( p_5(b, z_1, z_2) \)

\[ \rho_1 : p_1(X) \rightarrow \exists Y p_4(X, Y) \]
\[ \rho_2 : p_4(X, Y) \rightarrow \exists Z \exists W p_5(X, Y, Z), p_6(Y, W, X) \]
\[ \rho_3 : p_5(X, Y, Z), p_8(X, Y) \rightarrow \exists W p_9(Y, W) \]
\[ \rho_4 : p_5(X, Y, Z), p_6(Y, W, X) \rightarrow p_7(X, Y, W, Z) \]
\[ \rho_5 : p_7(X, Y, Z, W), p_3(V, X) \rightarrow p_8(X, Y) \]
The Tamed Case - Difficulty I

Active type of an atom

Lemma: Due to stickiness we can focus on a finite part of size $\#\text{pred} \cdot (\text{maxarity} + 1)^{\text{maxarity}}$. 
The Tamed Case - Difficulty II

Order of null generation - mixing of incompatible guarded nulls

$p_1(b) \rightarrow p_4(b, z_1) \rightarrow p_5(b, z_1, z_2)$
$p_6(z_1, z_3, b) \rightarrow p_7(b, z_1, z_3, z_2) \rightarrow p_8(b, z_1) \rightarrow p_9(z_1, z_4)$

$p_3(c, b) \rightarrow p_7(b, z_1, z_3, z_2)$

$p_5(X, Y, Z), p_6(Y, W, X) \rightarrow p_7(X, Y, W, Z)$

sticky (non-guarded) rule
The Tamed Case - Difficulty II

Order of null generation - mixing of incompatible guarded nulls

$p_1(b)$
$p_4(b, z_1)$
$p_5(b, z_1, z_2)$
$p_6(z_1, z_3, b)$
$p_7(b, z_1, z_3, z_2)$
$p_8(b, z_1)$
$p_9(z_1, z_4)$

$p_3(c, b)$

$p_5(X, Y, Z)$, $p_6(Y, W, X) \rightarrow p_7(X, Y, W, Z)$

sticky (non-guarded) rule

...mixing of guarded and sticky nulls
The Tamed Case - Difficulty II

Backward resolution steps (sticky steps)

\[ p_5(b, z_1, z_2) \]

\[ p_6(b, z_1, z_3) \]

\[ p_7(b, z_1, z_3, z_2) \]

\[ p_5(X, Y, Z), p_6(Y, W, X) \rightarrow p_7(X, Y, W, Z) \]
The Tamed Case

A (hybrid) alternating algorithm

• Guess the image of the given query - \( q(z_1, z_2, c, z_3), p(z_1, z_3, z_4) \)
• Guess a partition of nulls into sticky and guarded
• Guess the order of guarded null generation - \( z_1 \prec z_2 \quad z_1 \prec z_3 \prec z_4 \)
The Tamed Case

A (hybrid) alternating algorithm

- Guess the image of the given query \(- q(z_1, z_2, c, z_3), p(z_1, z_3, z_4)\)
- Guess a partition of nulls into sticky and guarded
- Guess the order of guarded null generation \(- z_1 \prec z_2 \prec z_1 \prec z_3 \prec z_4\)
- Universally prove the relevant chase derivations (guarded + sticky steps)

\[
\begin{align*}
|\text{atype}(\alpha, D, O)| & \leq \#\text{pred} \cdot (\text{maxarity} + 1)^{\text{maxarity}} \\
\Downarrowおる \uparrow
\end{align*}
\]

**Theorem:** Query answering under tame $G\mid S$ is in:

- $\text{AEXPSPACE} = 2\text{EXPTIME}$ (combined)
- $\text{APSPACE} = \text{EXPTIME}$ (bounded arity)
- $\text{ALOGSPACE} = \text{PTIME}$ (data).
Overview

- **Glut-guardedness** - guard only harmful variables
  [Krötzsch & Rudolph, IJCAI 2011]

- **Weak-stickiness** - only harmful join-variables stick to the inferred atoms
  [Cali, Gottlob & P., Artificial Intelligence 2010]

- **Tameness** - sticky rules do not feed the guard atom
  [this work]
Next Step: Datalog Rewriting

Towards practical query answering algorithms
Thank you!