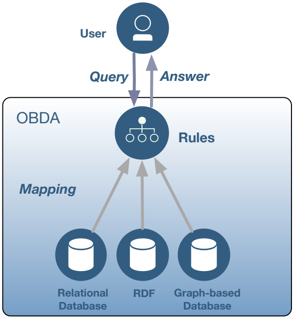
Querying Clique Guarded Existential Rules

Sen Zheng, Renate A. Schmidt

University of Manchester

November 15, 2019

Ontology Based Data Access



Ontology-based data access (OBDA) systems

- integrate schemas of heterogeneous databases
- present rules in decidable logics, e.g., description logics, guarded logics
- mainly concern answering queries

Context

Answering Boolean conjunctive query (BCQ)

- Returns an yes/no to a query
- Query containment/equivalence/evaluation
- Widely studied in literatures, e.g., [2, 3, 4, 5, 7, 8]

Clique guarded existential rules (CGER):

- Decidable
- Subsume Horn (loosely) guarded fragment and Horn-$\mathcal{ALCHOI}$
- No resolution-based approach deciding/querying CGER yet

Context

Answering Boolean conjunctive query (BCQ)

- Returns an yes/no to a query
- Query containment/equivalence/evaluation
- Widely studied in literatures, e.g., [2, 3, 4, 5, 7, 8]

Clique guarded existential rules (CGER):

- Decidable
- Subsume Horn (loosely) guarded fragment and Horn-$\mathcal{ALCHOI}$
- No resolution-based approach deciding/querying CGER yet

Aim:
Give a BCQ q, a set of CGERs Σ and a set of ground facts $\mathcal{D}$, check whether $\Sigma \cup \mathcal{D} \models q$.

BCQ and CGER

BCQ: $q = \exists X \phi(X)$
ϕ is a conjunction of atoms containing only variables and constants.

BCQ and CGER

BCQ: $q = \exists X \phi(X)$
ϕ is a conjunction of atoms containing only variables and constants.

$\exists xyz\ A(x, y) \wedge B(y, z)$

BCQ and CGER

BCQ: $q = \exists X \phi(X)$
ϕ is a conjunction of atoms containing only variables and constants.

$\exists xyz \; A(x, y) \wedge B(y, z)$

CGER: $\forall X \forall Y \phi(X, Y) \rightarrow \exists Z \psi(X, Z)$

- $\phi(X, Y)$ and $\psi(X, Z)$ are conjunctions of atoms
- X is not an empty set in $\exists Z \psi(X, Z)$
- each pair of free variables of $\exists Z \psi(X, Z)$ co-occur in at least one atom of $\phi(X, Y)$

BCQ and CGER

BCQ: $q = \exists X \phi(X)$
ϕ is a conjunction of atoms containing only variables and constants.

$\exists xyz\ A(x, y) \wedge B(y, z)$

CGER: $\forall X \forall Y \phi(X, Y) \rightarrow \exists Z \psi(X, Z)$

- $\phi(X, Y)$ and $\psi(X, Z)$ are conjunctions of atoms
- X is not an empty set in $\exists Z \psi(X, Z)$
- each pair of free variables of $\exists Z \psi(X, Z)$ co-occur in at least one atom of $\phi(X, Y)$

$\forall xyzv_1 v_2 v_3 (A_1(x, y, v_1) \wedge A_2(x, z, v_2) \wedge A_3(y, z, v_3) \rightarrow \exists w B(x, y, z, w))$

Decision Procedure

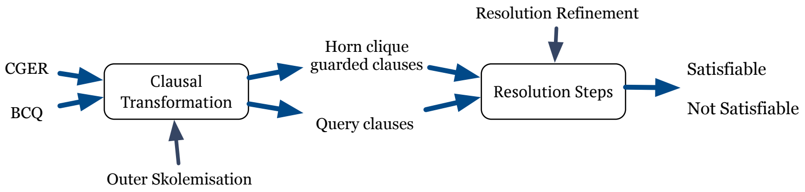


Figure: Procedure Overview

Three major steps to decide $\Sigma \cup \mathcal{D} \cup \neg q$:

- Clausal transformation
- Clause definement
- Resolution refinement

Clausal Transformation

1. Negating BCQ to obtain query clauses:

$\exists xyz\ A(x, y) \wedge B(y, z)$ becomes $\neg A(x, y) \vee \neg B(y, z)$

Clausal Transformation

1. Negating BCQ to obtain query clauses:

$\exists xyz\ A(x, y) \land B(y, z)$ becomes $\neg A(x, y) \lor \neg B(y, z)$

2. Using prenex normal form and outer Skolemisation to obtain Horn clique guarded clause (HCGC):

$\forall xyzv_1 v_2 v_3 (A_1(x, y, v_1) \land A_2(x, z, v_2) \land A_3(y, z, v_3) \rightarrow \exists w B(x, y, z, w))$

becomes

$\neg A_1(x, y, v_1) \lor \neg A_2(x, z, v_2) \lor \neg A_3(y, z, v_3) \lor B(x, y, z, fxyzv_1 v_2 v_3)$ where f is a Skolem function.

Query Clause and HCGC

Query clause: A negative clause containing only variables and constants.
$\neg A(x, y) \vee \neg B(y, z, a)$

Query Clause and HCGC

Query clause: A negative clause containing only variables and constants.
$\neg A(x, y) \vee \neg B(y, z, a)$

Horn clique guarded clause C:

- condensed Horn clause
- simple
  $\rightarrow$ no nested non-ground compound terms
- weakly covering
  $\rightarrow$ $var(t) = var(C)$ for any non-ground compound term t
- there is a clique in some negative flat literals (aka guards)
  $\rightarrow$ In C_1, a clique $\{x, y, z\}$ co-occur with each other in one of guards
- variables in C all occur in guards $\mathcal{G}$
  $\rightarrow$ $var(\mathcal{G}) = var(C)$

$$C_1 = \neg A_1(x, y, v_1) \vee \neg A_2(x, z, v_2) \vee \neg A_3(y, z, v_3) \vee B(x, y, z, fxyzv_1v_2v_3)$$

Loss of Weakly Covering Property

Example 1:

Given a set of HCGC C_1, C_2 and a query clause Q:

$$Q = \neg A_1(x, y) \vee \neg A_2(y, z)$$
$$C_1 = A_1(fx_1y_1, x_1) \vee \neg G_1(x_1, y_1)$$
$$C_2 = A_2(gx_2y_2, x_2) \vee \neg G_2(x_2, y_2)$$

Loss of Weakly Covering Property

Example 1:

Given a set of HCGC C_1, C_2 and a query clause Q:

$$Q = \neg A_1(x, y) \vee \neg A_2(y, z)$$
$$C_1 = A_1(fx_1y_1, x_1) \vee \neg G_1(x_1, y_1)$$
$$C_2 = A_2(gx_2y_2, x_2) \vee \neg G_2(x_2, y_2)$$

Perform resolution on Q, C_1, C_2:

the mgu is $\{x \mapsto f(gx_2y_2, y_1),\ x_1, y \mapsto gx_2y_2,\ z \mapsto x_2\}$,
the resolvent is $\neg G_1(g(x_2y_2), y_1) \vee \neg G_2(x_2, y_2)$.

Loss of Weakly Covering Property

Example 1:

Given a set of HCGC C_1, C_2 and a query clause Q:

$$Q = \neg A_1(x, y) \vee \neg A_2(y, z)$$
$$C_1 = A_1(fx_1y_1, x_1) \vee \neg G_1(x_1, y_1)$$
$$C_2 = A_2(gx_2y_2, x_2) \vee \neg G_2(x_2, y_2)$$

Perform resolution on Q, C_1, C_2:

the mgu is $\{x \mapsto f(gx_2y_2, y_1),\ x_1, y \mapsto gx_2y_2,\ z \mapsto x_2\}$,
the resolvent is $\neg G_1(g(x_2y_2), y_1) \vee \neg G_2(x_2, y_2)$.

Neither a query clause nor a HCGC.

Term Depth Increase

Example 2:

Given a set of HCGC:

$$C = \neg A_1(x, y, v_1) \vee \neg A_2(y, z) \vee \neg A_3(z, x) \vee D(x, z)$$
$$C_1 = A_1(fx_1, x_1, x_1) \vee \neg G_1(x_1)$$
$$C_2 = A_2(gx_2, gx_2) \vee \neg G_2(x_2)$$
$$C_3 = A_3(x_3, fx_3) \vee \neg G_3(x_3)$$

Term Depth Increase

Example 2:

Given a set of HCGC:

$$C = \neg A_1(x, y, v_1) \vee \neg A_2(y, z) \vee \neg A_3(z, x) \vee D(x, z)$$
$$C_1 = A_1(fx_1, x_1, x_1) \vee \neg G_1(x_1)$$
$$C_2 = A_2(gx_2, gx_2) \vee \neg G_2(x_2)$$
$$C_3 = A_3(x_3, fx_3) \vee \neg G_3(x_3)$$

Perform resolution on C, C_1, C_2, C_3:

the mgu is $\{x \mapsto fgx_2,\ y, z, x_1, x_3, v_1 \mapsto gx_2\}$,
the resolvent is $\neg G_1(gx_2) \vee \neg G_2(x_2) \vee \neg G_3(gx_2) \vee D(fgx_2, gx_2)$.

Top Variable Technique

Inspired by 'MAXVAR' [6], the top variable technique is to find the potentially deepest term to resolve upon.

Top Variable Technique

Inspired by 'MAXVAR' [6], the top variable technique is to find the potentially deepest term to resolve upon.

Using Example 2:

1. Order variables in the main premise by variable depth of their substitution.
   $\rightarrow \{x \mapsto fgx_2, y \mapsto gx_2, z \mapsto gx_2\}$, hence $x >_v y =_v z$.

Top Variable Technique

Inspired by 'MAXVAR' [6], the top variable technique is to find the potentially deepest term to resolve upon.

Using Example 2:

1. Order variables in the main premise by variable depth of their substitution.
   $\rightarrow \{x \mapsto fgx_2, y \mapsto gx_2, z \mapsto gx_2\}$, hence $x >_v y =_v z$.
2. Find top variables. $\rightarrow x$

Top Variable Technique

Inspired by 'MAXVAR' [6], the top variable technique is to find the potentially deepest term to resolve upon.

Using Example 2:

1. Order variables in the main premise by variable depth of their substitution.
   $\rightarrow \{x \mapsto fgx_2, y \mapsto gx_2, z \mapsto gx_2\}$, hence $x >_v y =_v z$.
2. Find top variables. $\rightarrow x$
3. Select literals that contain top variables. $\rightarrow A_1$ and A_3

Top Variable Technique

Inspired by 'MAXVAR' [6], the top variable technique is to find the potentially deepest term to resolve upon.

Using Example 2:

1. Order variables in the main premise by variable depth of their substitution.
   $\rightarrow \{x \mapsto fgx_2, y \mapsto gx_2, z \mapsto gx_2\}$, hence $x >_v y =_v z$.
2. Find top variables. $\rightarrow x$
3. Select literals that contain top variables. $\rightarrow A_1$ and A_3
4. Apply resolution on selected literal.
   $\rightarrow$ Applying resolution on C, C_1, C_3 derives
   $C_4 = \neg G_1(x_1) \vee \neg G_3(x_1) \vee \neg A_2(x_1, x_1) \vee D(fx_1, x_1)$

Top Variable Technique

Inspired by 'MAXVAR' [6], the top variable technique is to find the potentially deepest term to resolve upon.

Using Example 2:

1. Order variables in the main premise by variable depth of their substitution.
   $\rightarrow \{x \mapsto fgx_2, y \mapsto gx_2, z \mapsto gx_2\}$, hence $x >_v y =_v z$.
2. Find top variables. $\rightarrow x$
3. Select literals that contain top variables. $\rightarrow A_1$ and A_3
4. Apply resolution on selected literal.
   $\rightarrow$ Applying resolution on C, C_1, C_3 derives
   $C_4 = \neg G_1(x_1) \vee \neg G_3(x_1) \vee \neg A_2(x_1, x_1) \vee D(fx_1, x_1)$

Further application of resolution between C_2 and C_4 can be avoided using orderings.

Resolution Refinement *Query-Refine*

Within framework of [1], we use admissible orderings and (top) selection to guarantee termination.

- function symbols > constant symbols > predicate symbols

Resolution Refinement *Query-Refine*

Within framework of [1], we use admissible orderings and (top) selection to guarantee termination.

- function symbols $>$ constant symbols $>$ predicate symbols
- select at least one negative non-ground compound literals
  $\boxed{\neg G(gx)} \vee \neg A(x) \vee B(gx)$

Resolution Refinement *Query-Refine*

Within framework of [1], we use admissible orderings and (top) selection to guarantee termination.

- function symbols $>$ constant symbols $>$ predicate symbols
- select at least one negative non-ground compound literals
  $\boxed{\neg G(gx)} \vee \neg A(x) \vee B(gx)$
- Else find the maximal literal
  $\neg A(x) \vee B(gx)^*$

Resolution Refinement *Query-Refine*

Within framework of [1], we use admissible orderings and (top) selection to guarantee termination.

- function symbols $>$ constant symbols $>$ predicate symbols
- select at least one negative non-ground compound literals
  $\boxed{\neg G(gx)} \vee \neg A(x) \vee B(gx)$
- Else find the maximal literal
  $\neg A(x) \vee B(gx)^*$
- Else select all the negative literals containing top variables
  $\boxed{\neg A(x, y, v_1)} \vee \neg B(y, z) \vee \boxed{\neg C(z, x)}$ if x is a top variable.

Claims

Query-Res denotes the combination of:

- condensation
- tautology elimination
- ordered factoring defined by *Query-Refine*
- ordered resolution with selection defined by *Query-Refine*

Claim 1

Query-Res decides HCGCs and query clauses.

Claim 2

The combination of the previous clausal transformations and Query-Res decide the BCQ answering for CGERs.

Conclusion and Ongoing Work

We developed a decision procedure for answering Boolean conjunctive queries against clique guarded existential rules, based on ordered resolution and a sophisticated form of selection.

The procedure is sound and complete.

Conclusion and Ongoing Work

We developed a decision procedure for answering Boolean conjunctive queries against clique guarded existential rules, based on ordered resolution and a sophisticated form of selection.

The procedure is sound and complete.

Next step:

- Support our claims with formal proofs.
- Using TPTP as a benchmark to run experiments.

Conclusion and Ongoing Work

We developed a decision procedure for answering Boolean conjunctive queries against clique guarded existential rules, based on ordered resolution and a sophisticated form of selection.

The procedure is sound and complete.

Next step:

- Support our claims with formal proofs.
- Using TPTP as a benchmark to run experiments.

TPTP contains 92 HCGC problems, among which 66 are non-propositional.

Among those 66 problems, 65 are Horn guarded clause, 1 is Horn loosely guarded but not guarded. No problem is HGGC but not a Horn loosely guarded clause.

Leo Bachmair and Harald Ganzinger.
Resolution theorem proving.
In Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, pages 19–99. Elsevier and MIT Press, 2001.

Jean-Franois Baget, Michel Leclre, Marie-Laure Mugnier, and Eric Salvat.
On rules with existential variables: Walking the decidability line.
Artificial Intelligence, 175(9):1620–1654, 2011.

Vince Bárány, Georg Gottlob, and Martin Otto.
Querying the guarded fragment.
In *Proc. LICS'10*, pages 1–10. IEEE Computer Society, 2010.

Andrea Calì, Georg Gottlob, and Michael Kifer.
Taming the infinite chase: Query answering under expressive relational constraints.
J. Artif. Int. Res., 48(1):115–174, 2013.

Andrea Calì, Georg Gottlob, and Thomas Lukasiewicz.

A general datalog-based framework for tractable query answering over ontologies.
Journal of Web Semantics, 14:57 – 83, 2012.

Hans de Nivelle and Maarten de Rijke.
Deciding the guarded fragments by resolution.
J. Symb. Comput., 35(1):21–58, 2003.

Birte Glimm, Carsten Lutz, Ian Horrocks, and Ulrike Sattler.
Conjunctive query answering for the description logic shiq.
J. Artif. Int. Res., 31(1):157–204, 2008.

Georg Gottlob, Andreas Pieris, and Lidia Tendera.
Querying the guarded fragment with transitivity.
In Fedor V. Fomin, Rūsiņš Freivalds, Marta Kwiatkowska, and David Peleg, editors, *Automata, Languages, and Programming*, pages 287–298. Springer, 2013.