Monadic Datalog Containment

Michael Benedikt Pierre Bourhis Pierre Senellart

University of Oxford and Télécom ParisTech

ICALP 2012, 13 July

Datalog

Basic query language with recursion.

$$\mathbf{ReachGood}() \leftarrow \mathit{Start}(x), \mathit{Reach}(x,y), \mathit{Good}(y)$$
$$\mathit{Reach}(x,y) \leftarrow \mathit{Reach}(x,z), \mathit{Reach}(z,y)$$
$$\mathit{Reach}(x,y) \leftarrow E(x,y)$$

- Rules consisting of Horn clauses.
- Heads of rules are *intensional* predicates
- Other predicates are extensional (input) predicates
- Distinguished **goal** predicate

Evaluation over an instance : least fix point semantics

Datalog

Basic query language with recursion.

$$\mathbf{ReachGood}() \leftarrow \mathit{Start}(x), \mathit{Reach}(x,y), \mathit{Good}(y)$$
$$\mathit{Reach}(x,y) \leftarrow \mathit{Reach}(x,z), \mathit{Reach}(z,y)$$
$$\mathit{Reach}(x,y) \leftarrow E(x,y)$$

- Rules consisting of Horn clauses.
- Heads of rules are *intensional* predicates
- Other predicates are extensional (input) predicates
- Distinguished **goal** predicate

Evaluation over an instance : least fix point semantics

Monadic Datalog (MDL) : all intensional predicates are unary.

Containment of Datalog

$Q \subseteq Q'$ iff for every input instance D, $Q(D) \subseteq Q'(D)$

Bad news [Shmueli, 1987]

Datalog containment and equivalence are undecidable

Containment of Datalog

$Q \subseteq Q'$ iff for every input instance D, $Q(D) \subseteq Q'(D)$

Bad news [Shmueli, 1987]

Datalog containment and equivalence are undecidable

Decidable fragments

[Cosmadakis et al., 1988] Decidability of the containment of a MDL program in a MDL program is 2EXPTIME and EXPTIME-hard

[Chaudhuri and Vardi, 1992] Decidability of the containment of a DL program in an union of conjunctive queries (UCQ) is 2EXPTIME-complete

Containment of Datalog

$Q \subseteq Q'$ iff for every input instance D, $Q(D) \subseteq Q'(D)$

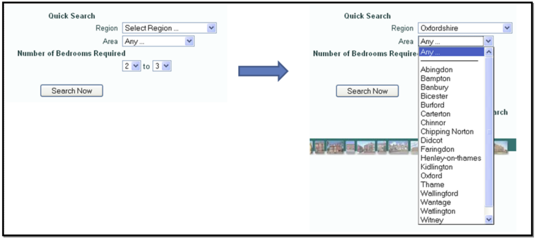
(formerly) Open Question

Tight bound on MDL into MDL/UCQ containment

Application : Querying with limited access

Limited access : Data accessible through Forms and Web Services

- Can not give a general query to the data, can only do lookups on particular fields
- Values used in lookups must either be known beforehand, or result from other lookups



Method ApartmentFind :
Region, Area, NumBeds $\rightarrow$ Address, Price, Description, Link

Accessible data

- Schema = Relations + Access methods
- Access method m to a relation : set of input attributes $input(m)$ requiring known values
- An access to method m_i is a binding of the input positions of m_i. It returns a set of tuples.
- A valid access sequence : binding values appear in the previous outputs or are constants.
- Accessible data of D, $Acc(D)$: data of D extractable from valid access sequences

Containment under limited access

Q is included in Q' under limited access iff for any D,
$Q(Acc(D)) \subseteq Q'(Acc(D))$

(formerly) Open Question

What is the complexity of query equivalence, containment under access patterns ?

- Single access method per relation
- No initial seed values

Containment under limited access

Q is included in Q' under limited access iff for any D, $Q(Acc(D)) \subseteq Q'(Acc(D))$

Reduction to containment of MDL in UCQs

(i) Computation of the accessible values by a MDL program

$$Accessible(x_j) \leftarrow (R(\vec{x}) \wedge \bigwedge_{i \in input(m)} Accessible(x_i))$$

$$Accessible(c) \leftarrow$$

(ii) Restriction of Q to the accessible data : $Q_{Acc} \in MDL$.

(iii) Containment of Q in Q' under limited access iff Q_{Acc} is contained in Q'.

.

(Formerly) Open Questions and Results

- What about containment under limited access patterns ?
  - Containment UCQs under limited access : 2EXPTIME-complete
  - Containment with a single method per relation : NEXPTIME-complete
  - Containment with a single method per relation and without seed values : EXPTIME-complete

(Formerly) Open Questions and Results

- What about containment under limited access patterns ?
  - Containment UCQs under limited access : 2EXPTIME-complete
  - Containment with a single method per relation : NEXPTIME-complete
  - Containment with a single method per relation and without seed values : EXPTIME-complete
- Is the 2EXPTIME bound on MDL/UCQ containment tight ?
  Containment of MDL/UCQ : 2-EXPTIME-Hard

MDL Containment in UCQs

[Chaudhuri and Vardi, 1992]

(M)DL containment in UCQs is in 2EXPTIME ;

MDL Containment in UCQs

[Chaudhuri and Vardi, 1992]
(M)DL containment in UCQs is in 2EXPTIME ;

New Proof

- Q is not contained in Q' iff there is a witness instance in which Q holds and Q' does not hold.
- The witness instance can be taken to be tree-like – of tree-width at most $|Q|$.
  - Tree like instances of Q represented by a tree automaton T_Q exponential in $|Q|$.
- Tree-like instances not satisfying Q' can be described by a tree automaton $T_{\neg Q'}$ double exponential in $|Q|$ and in $|Q'|$

Containment of Q in Q' iff $T_Q \cap T_{\neg Q'}$ is empty

In more detail

Instance I is k-very tree-like :

- Tree decomposition of tree width k
- Only at most one value shared between nodes
- Two nodes sharing a value are parent/child

Lemma

Q is not contained in Q' iff there exists a $|Q|$-very-tree-like instance satisfying Q and not Q'

In more detail

How to build $T_{\neg Q'}$

- Types of tree-like nodes n : The set of maximal connected subqueries of Q' satisfied by the atoms in the bags of the subtree rooted at n
- State of a node in $T_{\neg Q'}$ = Type of the node
  - The type of a node is determined by the types of its children and the facts of the node

In more detail

How to build $T_{\neg Q'}$

- Types of tree-like nodes n : The set of maximal connected subqueries of Q' satisfied by the atoms in the bags of the subtree rooted at n
- State of a node in $T_{\neg Q'}$ = Type of the node
  - The type of a node is determined by the types of its children and the facts of the node

Lemma

The set of k very-tree-like instances not satisfying Q' is represented by a tree automaton polynomial in the number of types and exponential in k

How to improve the bound on the number of types?

Unique Mapping Condition (UMC) of I

for any conjunctive query Q', any atom A, any node n of I there exists at most one maximal connected subquery Q'' of Q' mapping into the subtree of n such that A maps into a fact of n.

How to improve the bound on the number of types?

Unique Mapping Condition (UMC) of I

for any conjunctive query Q', any atom A, any node n of I there exists at most one maximal connected subquery Q'' of Q' mapping into the subtree of n such that A maps into a fact of n.

Theorem

If a regular class C of tree-like instances satisfies the UMC, then there is an EXPTIME function taking a (U)CQ Q' to a tree automaton that accepts the instances of C not satisfying Q'.

Thus containment of a tree automaton in UCQ over trees in a such class is EXPTIME

Tree decompositions with UMC

A diversified tree-like instance is a very tree-like instance in which :

(i) for each node n other than the root, there are not two facts in n having the same relation name

(ii) a value shared by two nodes cannot appear in the same position in two facts having the same relation name.

Key Lemmas

(i) The class of diversified tree-like instances satisfies the UMC.

(ii) The following containment problems always have diversified tree-like counter-examples

- Containment of UCQ with limited access with a single access per relation
- Containment of Monadic datalog where a relation symbol appears in at most one rule in UCQ

Putting it all together

- Counterexamples to containment for limited-access schemas with a single access per relation can be taken to be diversified tree-like.
- Diversified instances satisfy the UMC, so they have unique maximal subqueries.
- We can construct a tree automaton that captures all diversified tree-like counterexamples, whose types are the vectors of maximal queries.
- Creating this automaton and checking non-emptiness can be done in EXPTIME.

Other Upper Bounds

Upper bounds : use analysis of tree-like models. Also can be used to show :

- Results in the presence of constants – still have exponential sized counterexample models, but complexity moves to NEXPTIME from EXPTIME
- New results over trees – tree automata containment in a tree pattern with only child axis is EXPTIME

Why MDL/UCQ Containment is 2EXPTIME-hard ?

Idea : Reduction from the problem for validity of tree automaton over conjunctive queries with descendant relations [Björklund et al., 2008].

Summary

Containment of MDL in UCQs

- 2EXPTIME-hardness
- Decidability of containment hinges on sufficiency of tree-like models.
- Complexity related to "fine structure" of tree-like models, which can be related to syntactic restrictions on the MDL query.

Summary

Containment of MDL in UCQs

- 2EXPTIME-hardness
- Decidability of containment hinges on sufficiency of tree-like models.
- Complexity related to "fine structure" of tree-like models, which can be related to syntactic restrictions on the MDL query.

Current and Future Work

- Containment of MDL with constraints or over particular structures (words, trees).

Henrik Björklund, Wim Martens, and Thomas Schwentick. Optimizing conjunctive queries over trees using schema information. In MFCS, 2008.

Surajit Chaudhuri and Moshe Y. Vardi. On the equivalence of recursive and nonrecursive Datalog programs. In PODS, 1992.

Stavros Cosmadakis, Haim Gaifman, Paris Kanellakis, and Moshe Vardi. "Decidable Optimization Problems for Database Logic Programs". In Proceedings of the 20th Annual ACM Symposium on Theory of Computing, pages 477–490, 1988.

Oded Shmueli. Decidability and Expressiveness Aspects of Logic Queries. In Proceedings of the 6th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'87), pages 237–249, 1987.