Query Containment for Conjunctive Queries with Safe Negation

Fang Wei and Georg Lausen
University of Freiburg,Germany
{fwei,lausen}@informatik.uni-freiburg.de

Abstract

The problem of query containment for conjunctive queries with safe negated subgoals has drawn considerably less attention in the past than other variants of the containment problem of conjunctive queries. In [LS93] uniform containment is discussed, which is a sufficient, however not necessary condition for containment. In [Ull97] it is argued that the complexity of the containment test is Π_2^p-complete, however the given arguments cannot be used to design a general algorithm. In this paper, a general algorithm for testing the containment of conjunctive queries with safe negated subgoals and built-in predicates is presented. To any conjunctive query with safe negation and built-in predicates we construct an equivalent set of subqueries, where each subquery is in a certain normal form. The containment problem for conjunctive queries obeying the normal form can be polynomially reduced to the containment problem of conjunctive queries without negated subgoals, however with built-in predicates. The complexity of the latter problem is known to be Π_2^p-complete. Testing containment for conjunctive queries with safe negated subgoals and built-in predicates thus has the same complexity as testing containment for its positive counterparts, whenever the given queries are in normal form. For an arbitrary query, however, our approach may require an exponential number of tests, in the worst case. In addition it is shown that known results from the literature can be applied to lower the complexity by applying certain restrictions to the queries.

1 Introduction

This paper considers the problem of query containment of conjunctive queries (CQ) with safe negated subgoals $CQ^{\neg}$. The query containment problem is to check whether the answer set of one query is always a subset of another query for any given database. Algorithms for query containment are of interest in several contexts in the database area. For query optimization [CM77, SY80] the task is to find redundant subgoals. Algorithms for data integration on the web rely on query rewriting using views, for which deciding query containment is a crucial sub-step [LMSS95]. Furthermore, query containment has also been used for checking integrity constraints [GSUW94], and for deciding query independence of updates [LS93].

Starting with the NP-completeness result proposed by Chandra and Merlin [CM77], many researchers have been working on extensions of the containment question. Containment of CQ's with inequalities is discussed in [Klu88, ZÖ97]. Containment of unions of CQ's is treated in [SY80], containment of CQ's with negated subgoals in [LS93, Ull97], containment over complex objects in [LS97], and over semi-structured data with regular expression in [FLS98].

The problem of query containment for conjunctive queries with safe negated subgoals has drawn considerably less attention in the past. In [LS93] uniform containment is discussed, which is a sufficient, however not necessary condition for containment. In [Ull97] it is argued that the complexity of the containment test is Π_2^p. The algorithmic idea behind is based on the approach of canonical databases [RSUV89], where an exponential number of *canonical* databases must be tested. The following example explains the approach.

Example 1. Consider the following queries of Q_1 and Q_2 over the predicate $a/2$:

$$\begin{array}{lll} Q_1: & p(X,Z) & \text{:-} & a(X,Y), a(Y,Z), \neg a(X,Z). \\ Q_2: & p(A,C) & \text{:-} & a(A,B), a(B,C), \neg a(A,D). \end{array}$$

□

In order to show $Q_1 \subseteq Q_2$, [Ull97] considers all 5 partitions of $\{X, Y, Z\}$; all variables in one set of a certain partition are replaced by the same constant. From each partition, a canonical database built out of the positive subgoals is generated according to the predicates in the body of Q_1. It is shown, that whenever the body of Q_1 can be satisfied with respect to a canonical database, that then both Q_1 and Q_2 will yield the same answers. From this it can be concluded that $Q_1 \subseteq Q_2$, which is actually true in this case.

However, Example 2 shows that this method of partitioning variables will in general not yield the desired results.

Example 2. Consider the following queries Q_1 and Q_2:

$$\begin{array}{llll} Q_1: & q(X) & \text{:-} & p(X), \neg r(X). \\ Q_2: & q(X) & \text{:-} & p(X), \neg t(X). \end{array}$$

There is only one variable X in Q_1, so the only partition would be to assign a constant to X, say 0, and to generate a canonical database $\{p(0)\}$. Because the fact $r(0)$ is not in the database, we get the tuple $q(0)$ from the head of Q_1. Next we apply the canonical database in Q_2, while $t(0)$ is not in the database either, we get $q(0)$ from Q_2 too. So we have to conclude that $Q_1 \subseteq Q_2$, which is, obviously not the case. Take a simple database $\{p(0), t(0)\}$, then we will get the tuple $q(0)$ from the query Q_1 but the empty set from Q_2, which gives us a counter-example. □

In this paper, we propose a new method to solve the general query containment problem for conjunctive queries with negated subgoals and built-in predicates ($CQ^{\neg}$). Let Q_1, Q_2 be two such queries and let $Q_1 \subseteq Q_2$ be the containment to be tested. Whenever Q_1 obeys a certain normal form, the containment problem can be polynomially reduced to the containment problem of conjunctive queries without negated subgoals, however with built-in predicates. The complexity of the latter problem is known to be Π_2^p-complete. Testing containment for conjunctive queries with safe negated subgoals and built-in predicates thus has the same complexity of testing containment for its positive counterparts, whenever the given query Q_1 is in normal form. Whenever Q_1 does not obey the required normal form, we construct for Q_1 an equivalent set of subqueries, where each subquery does obey the normal form. As the number of subqueries may be exponential in the length of Q_1, our approach in this situation may require an exponential number of tests, in the worst case.

In addition, we investigate to lower the complexity of containment checking for $CQ^{\neg}$'s by adding some syntactical restrictions to the queries. It is well-known that conjunctive queries containment problem $Q_1 \subseteq Q_2$ is polynomially solvable if each predicate occurs at most twice in the Q_1 [Sar91] and Kolaitis [KMT98] proved that the following problem is in NP: given two safe conjunctive $\{\neq\}$ queries Q_1, Q_2, and Q_1 is a fixed conjunctive $\{\neq\}$ query, is $Q_1 \subseteq Q_2$? We show for *general* $CQ^\neg$'s, i.e. the normal form needs not necessarily be fulfilled, that the containment problem can be polynomially solved if every positive database predicate occurs at most twice in the body of Q_1 and each negated predicate does not occur positive; the containment checking problem of $CQ^\neg$'s without built-in predicates: $Q_1 \subseteq Q_2$ is in NP if Q_1 is a fixed query, that is, Q_1 is not any more part of the input to the algorithmic problem considered. This keeps to be the case also for the $CQ^\neg$'s with built-in predicates.

2 Preliminaries

2.1 Query Containment

A *conjunctive query with negation* ($CQ^\neg$) extends usual CQ by allowing negated subgoals in the body. It has the following form:

$$h \quad \text{:-} \quad p_1, ..., p_n, \neg s_1, ..., \neg s_m.$$

where $h, p_1, ..., p_n, s_1, ..., s_m$ are predicates whose arguments are variables or constants, h is the *head*, $p_1, ..., p_n$ are the positive subgoals, and $s_1, ..., s_m$ are the negated subgoals. We assume that (1) the variables appearing in the head also appear in the body; (2) all the variables appearing in the negated subgoals also appear in positive ones, which is also called the *safeness* condition for $CQ^\neg$. The examples in this paper are *safe* $CQ^\neg$s if not mentioned otherwise.

A $CQ^\neg$ is applied to a set of finite database relations by considering all possible substitutions of values for the variables in the body. If a substitution makes all the positive subgoals true and all the negated subgoals false (i.e. they do not exist in the database), then the same substitution, applied to the head, composes one answer of the conjunctive query. The set of all answers with respect to a certain database D is denoted by $Q(D)$.

The containment of $CQ^\neg$'s is defined in the same as for normal conjunctive queries: a $CQ^\neg$ Q_1 is contained in another one Q_2, denoted as $Q_1 \subseteq Q_2$, if for all databases D, $Q_1(D) \subseteq Q_2(D)$. Two $CQ^\neg$'s are equivalent if and only if each is contained in the other.

When built-in predicates are allowed, we have the following form:

$$h \quad \text{:-} \quad p_1, ..., p_n, \neg s_1, ..., \neg s_m, c_Q.$$

where h and $p_1, ...p_n, s_1, ..., s_m$ are defined as before and c_Q is a conjunction of built-in predicates. We refer to c_Q also as the *initial* constraints of the query. ¿From now on, we use $CQ^\neg$ for a conjunctive query with safe negated subgoals *and* built-in predicates, if not explicitly stated otherwise.

2.2 The Containment Checking Algorithm for CQ's with Built-in Predicates

An algorithm for checking the containment of CQ's with built-in predicates has been proposed in [ZÖ97]. Let a *containment mapping* be a mapping from a set of variables to a set of variables and constants.

Definition 1 (Containment for CQ's with built-in predicates [ZÖ97]). *Consider two CQ's Q_1 and Q_2 with built-in predicates:*

$$\begin{aligned} Q_1 : h &\ \text{:-}\ \ p_1, ..., p_n, c_{Q1}. \\ Q_2 : h &\ \text{:-}\ \ q_1, ..., q_l, c_{Q2}. \end{aligned}$$

Then $Q_1 \subseteq Q_2$ if and only if there are containment mappings $\rho_1, ..., \rho_r$ from the subgoals in Q_2 to those in Q_1, such that c_{Q1} implies $\rho_1(c_{Q2}) \vee ... \vee \rho_r(c_{Q2})$. □

Example 3. Consider queries Q_1 and Q_2:

$$\begin{aligned} Q_1 : q(U, V) &\ \text{:-}\ \ r(U, V), r(V, U). \\ Q_2 : q(X, Y) &\ \text{:-}\ \ r(X, Y), X \geq Y. \end{aligned}$$

There are two containment mappings from Q_2 to Q_1, namely $\rho_1 : \{X \to U, Y \to V\}$ and $\rho_2 : \{X \to V, Y \to U\}$. If we apply the containment mappings to the built-in predicates, we have $\texttt{true} \to U \geq V \vee V \geq U$. As a result, $Q_1 \subseteq Q_2$. □

3 Query Containment for $CQ^{\neg}$'s

The containment problem for conjunctive queries (without negation, without built-in predicates) is shown to be NP-complete [CM77]. When in addition built-in predicates are allowed, the complexity becomes Π_2^p-complete [vdM92].

3.1 Normal Form

Definition 2 (Normal Form for $CQ^{\neg}$). *Consider a $CQ^{\neg}$ with built-in predicates of the following form:*

$$Q : \quad h \ \ \text{:-}\ \ p_1, ..., p_n, \neg s_1, ..., \neg s_m, c_Q.$$

Q is said to be in normal form, *if the following condition is fulfilled. If there is one positive predicate p_i of the form $q(X_1, ..., X_k)$ and one negative predicate $\neg s_j$ of the form $\neg q(Y_1, ..., Y_k)$, then at least one inequality expression $X_i \neq Y_i$ $(1 \leq i \leq k)$ is in c_Q or can be inferred from c_Q.* □

Normal form in the above sense prohibits the appearance of variable bindings which satisfy c_Q, however will yield a fact and its negation thus giving rise to an unsatisfiable body of the query. To each $CQ^{\neg}$ we can derive an equivalent normal form. However, to get equivalence, we have to consider a union of subqueries, each being in normal form. In the worst case, we may get exponentially many subqueries. The distinct subqueries forming the equivalent set of subqueries in normal form can be derived by adding inequalities to the original query. For example, if only $q(X_1, ..., X_k)$ and $\neg q(Y_1, ..., Y_k)$ fulfill the condition of definition 2, then there are k possible ways to add inequalities to the query to bringing it in normal form. Assume a predicate p of arity n appears positively k times and negated l times, then there may exist n^{kl} possible ways to derive a subquery in normal form. Note, that to get the subqueries in normal form, we leave the subgoals and the initial constraints of the original query unchanged. The newly introduced inequalities form the *normal* constraints of the subquery denoted $\bar{c}_Q$.

Proposition 1. *Each $CQ^{\neg}$ Q is equivalent to a union of $CQ^{\neg}$'s in normal form.* □

Example 4. Consider a $CQ^{\neg}$ Q together with its equivalent set of subqueries in normal form $\{Q_1, Q_2\}$:

$$\begin{array}{lll} Q: & q(X,Z) & \text{:-} \quad a(X,Y), a(Y,Z), \neg a(Z,X). \\ Q_1: & q(X,Z) & \text{:-} \quad a(X,Y), a(Y,Z), \neg a(Z,X), X \neq Z. \\ Q_2: & q(X,Z) & \text{:-} \quad a(X,Y), a(Y,Z), \neg a(Z,X), X \neq Y, Y \neq Z. \end{array}$$

There are no initial constraints in Q. Consequently, Q_1 has normal constraint $X \neq Y$ and Q_2 has normal constraints $X \neq Y, Y \neq Z$. We have $Q \equiv Q_1 \cup Q_2$. Note that because the variables in the occurrences of a are not distinct, we need only 2 subqueries, instead of the possible 4. □

3.2 A Necessary Condition for Containment

Now we are going to link the containment of $CQ^{\neg}$ in normal form (with safe negation and built predicates) to the containment of CQ's (without negation, however with built-in predicates). To simplify notation we will not further distinguish between initial and normal constraints; we refer to both by c_Q, in the sequel, where Q is the query under consideration.

Definition 3 (Super-Positive SP Q^+). *Given a $CQ^{\neg}$ Q:*

$$Q: \quad h \quad \text{:-} \quad p_1, ..., p_n, \neg s_1, ..., \neg s_m, c_Q.$$

The SP version of Q, Q^+, is defined as follows:

$$Q^+: \quad h \quad \text{:-} \quad p_1, ..., p_n, c_Q.$$

□

Compared with $CQ^{\neg}$ Q, the query Q^+ has simply the same query head but only the positive subgoals in the body. Because Q^+ has no negation in the body, it has the monotonic properties of normal CQ's. However, it is interesting to notice that although any $CQ^{\neg}$ is equivalent to a union of $CQ^{\neg}$ in normal form, the corresponding SP's are not equivalent, in general. Taking example 4, it is obvious that $Q^+ \not\equiv Q_1^+ \cup Q_2^+$. The following lemmata provide the basis for the following theorem.

Lemma 1. *Given a $CQ^{\neg}$ Q with negated subgoals and its SP Q^+, then $Q \subseteq Q^+$.* □

Lemma 2. *Let Q be a $CQ^{\neg}$ in normal form and Q^+ its SP, D a database, and a tuple $t \in Q^+(D)$ but $t \notin Q(D)$. Then there exists a database $D' \subseteq D$, such that $t \in Q(D')$.*

Proof. Assume that Q and Q^+ have the forms of Definition 3. Because $t \in Q^+(D)$, let σ be any substitution for variables in Q^+ that yields t (note that the built-in predicates should be satisfied too). Since $t \notin Q(D)$, there is at least one s_i $(1 \leq i \leq m)$, such that $\sigma(s_i) \in D$. ¿From the restriction of normal form, $\sigma(s_i)$ is a distinct element from $\{\sigma(p_1), ..., \sigma(p_n)\}$, or $\sigma(s_i) \notin \{\sigma(p_1), ..., \sigma(p_n)\}$. That is, if we remove $\sigma(s_i)$ from D, it is still hold that $t \in Q^+(D)$. In the same way we can remove all the facts $\sigma(s_i)$ $(1 \leq i \leq m)$ from D, and at the end we have both $t \in Q^+(D)$ and $t \in Q(D)$. □

Theorem 1. *Let Q_1 and Q_2 be two $CQ^{\neg}$'s and Q_1 in normal form; let Q_1^+ and Q_2^+ be their SP respectively. If $Q_1 \subseteq Q_2$, then $Q_1^+ \subseteq Q_2^+$.*

Proof. Assume $Q_1 \subseteq Q_2$ and $t \in Q_1^+(D)$ where D is any database. We show that $t \in Q_2^+(D)$.

- Case 1. If $t \in Q_1(D)$, then the situation is simple: we get $t \in Q_2(D)$, then from Lemma 1, we have $t \in Q_2^+(D)$.
- Case 2. Assume $t \notin Q_1(D)$. Because $t \in Q_1^+(D)$, from Lemma 2 we obtain a $D' \subseteq D$ where $t \in Q_1(D')$. Then $t \in Q_2(D')$ is obtained from $Q_1 \subseteq Q_2$. Similar to the Case 1, we have $t \in Q_2^+(D')$. Observe that here Q_2^+ is a CQ with only positive subgoals which is monotonic. Thus, $t \in Q_2^+(D)$ since $D' \subseteq D$.

□

Theorem 1 provides a necessary but not sufficient condition for query containment of CQ's with negated subgoals.

Example 5. Consider $CQ^{\neg}$'s Q_1 and Q_2 with Q_1 in normal form:

$$\begin{array}{lll} Q_1: & p(X,Y) & \text{:- } a(X,Z), a(Z,Y), \neg a(X,Y), X \neq Z, Y \neq Z. \\ Q_2: & p(A,B) & \text{:- } a(A,B), a(B,C), a(B,D), \neg a(A,D). \end{array}$$

We have $Q_1^+ \subseteq Q_2^+$. □

However, in the preceding example, to check whether Q_1 is actually contained in Q_2 (which is here the case), another condition is needed, which is introduced next.

3.3 Containment Checking

The next theorem constitutes the main result of this section.

Theorem 2. *Let Q_1 and Q_2 be two $CQ^{\neg}$, where Q_1 is in normal form. Let in addition be $Q_1^+ \subseteq Q_2^+$.*

$$\begin{array}{lll} Q_1: & h & \text{:- } p_1, ..., p_n, \neg s_1, ..., \neg s_m, c_{Q1}. \\ Q_2: & h & \text{:- } q_1, ..., q_l, \neg a_1, ..., \neg a_k, c_{Q2}. \end{array}$$

Then $Q_1 \subseteq Q_2$ if and only if

(1) there is a set of containment mappings $\rho_1, ..., \rho_r$ from Q_2 to Q_1, such that c_{Q1} implies $\rho_1(c_{Q2}) \vee ... \vee \rho_r(c_{Q2})$.

(2) for each $i, j (1 \leq i \leq r, 1 \leq j \leq k)$, the conjunctive query

$$Q'': \quad h \quad \text{:- } p_1, ..., p_n, \neg s_1, ..., \neg s_m, \rho_i(a_j).$$

is unsatisfiable.

Proof. $\Leftarrow$: Given $CQ^{\neg}$'s Q_1 and Q_2 and a database D. Let a tuple $t \in Q_1(D)$, and we must prove that $t \in Q_2(D)$.

Since $t \in Q_1(D)$, we have $t \in Q_1^+(D)$ and $t \in Q_2^+(D)$. Let σ be the substitution for all the variables in Q_1 that yields t. Because $t \in Q_2^+(D)$, there exists one $\rho_i (1 \leq i \leq r)$, such that the substitution from variables in Q_2^+ that yields t has the form $\{\sigma(\rho_i(q_1)), ..., \sigma(\rho_i(q_l))\} \subseteq D$. Since we assume for each $i, j (1 \leq i \leq r, 1 \leq j \leq k)$, that the conjunctive query Q'' is unsatisfiable, we have $t \notin Q''(D)$. From the definition of containment mapping, we have $\{\rho_i(q_1), ..., \rho_i(q_l)\} \subseteq \{p_1, ..., p_n\}$. As a result, we conclude that $t \notin Q_2^j(D) (1 \leq j \leq k)$, where each Q_2^j has the form:

$$Q_2^j: \quad h \quad \text{:-} \quad q_1, ..., q_l, a_j, c_{Q2}.$$

Together with the fact $t \in Q_2^+(D)$, we have for each $a_j (1 \leq j \leq k)$, $\sigma(\rho_i(a_j)) \notin D$. Consequently, we have $t \in Q_2(D)$.

$\Rightarrow$: Given that $Q_1 \subseteq Q_2$, it must be shown that there is a set of containment mappings $\rho_1, ..., \rho_r$ from Q_2 to Q_1, such that c_{Q1} implies $\rho_1(c_{Q2}) \vee ... \vee \rho_r(c_{Q2})$ and for each i, j $(1 \leq i \leq r, 1 \leq j \leq k)$, Q'' is unsatisfiable.

Assume that there are b sets of containment mappings from Q_2 to Q_1, such that for each i $(1 \leq i \leq b)$, there are r_i containment mappings, and c_{Q1} implies $\rho_{i,1}(c_{Q2}) \vee ... \vee \rho_{i,r_i}(c_{Q2})$, where $(1 \leq i \leq b)$; and for each set there are at least one $\rho_{i,v}$ and one a_j where$(1 \leq i \leq b, 1 \leq v \leq r_i, 1 \leq j \leq k)$, such that the query

$$Q'': \quad h \quad \text{:-} \quad p_1, ..., p_n, \neg s_1, ..., \neg s_m, \rho_{i,v}(a_j).$$

is satisfiable. Then we must show that $Q_1 \not\subseteq Q_2$.

We build up a *canonical database* D in the following way: *freeze* the body of Q_1 by mapping each variable in Q_1 to a distinct constant, and call the mapping σ. We add first $\sigma(p_1), ..., \sigma(p_n)$ to D. Since Q_1 is in normal form, $\sigma(s_i)$ $(1 \leq i \leq m) \notin D$. Next we add each $\sigma(\rho_{i,v}(a_j))$ in D, where $(1 \leq i \leq b, 1 \leq v \leq r_i, 1 \leq j \leq k)$. As a result, if we apply Q_2 to D, we have an empty answer set since for each substitution θ that make Q_2^+ true, there is at least one a_j $(1 \leq j \leq k)$, such that $\theta(a_j) \in D$. It is obvious that there exists a tuple t, such that $t \in Q_1(D)$. Finally we provide a contradiction of $Q_1 \subseteq Q_2$. □

Example 6. Given the queries Q_1 and Q_2:

$$\begin{array}{llll} Q_1: & h & \text{:-} & r(U,V), r(V,U), \neg t(U), \neg t(V). \\ Q_2: & h & \text{:-} & r(X,Y), X \geq Y, \neg t(X). \end{array}$$

Q_1 is in normal form, and there are containment mappings ρ_1 and ρ_2 as in Example 3. Next we construct the query Q''_{ρ_1} and Q''_{ρ_2}, by using $\rho_1(t(X)) = t(U)$ and $\rho_2(t(X)) = t(V)$:

$$\begin{array}{llll} Q''_{\rho_1}: & h & \text{:-} & r(U,V), r(V,U), \neg t(U), \neg t(V), t(U). \\ Q''_{\rho_2}: & h & \text{:-} & r(U,V), r(V,U), \neg t(U), \neg t(V), t(V). \end{array}$$

Both of the above queries are unsatisfiable; thus we have proved $Q_1 \subseteq Q_2$. □

As satisfiability of conjunctive queries can be tested in polynomial time [Ull89], testing containment according to Theorem 2 has the same complexity as testing containment of the positive counterparts, which is Π_2^p-complete.

Now let us consider the situation, where we have two arbitrary $CQ^{\neg}$'s Q_1 and Q_2. In order to test $Q_1 \subseteq Q_2$, Q_1 should be represented by a set of normal forms of it, which is, in the worst case, exponential in the size of the length of Q_1. However, it is relatively small comparing to the size of data. Moreover, if we fix the query Q_1, by taking only Q_2 as input to the algorithm, the size of the normal forms is not any more exponential. This will be discussed in the following subsection.

3.4 The Polynomial and NP Cases of $CQ^{\neg}$'s Containment

In view of the high intractability of the containment checking problem, (pure conjunctive queries with NP-complete, and Π_2^p-complete for CQ's with built-in predicates [vdM92]), researchers have discovered some lower complexity cases by restricting the query classes syntactically. Saraiya [Sar91] showed that pure conjunctive query containment $Q_1 \subseteq Q_2$ can be solved in linear time, if every database predicate occurs at most twice in the body of Q_1. Kolaitis [KMT98] proved that the following problem is in NP: given two safe conjunctive $\{\neq\}$ queries Q_1, Q_2, and Q_1 is a fixed conjunctive $\{\neq\}$ query, is $Q_1 \subseteq Q_2$? Interestingly, the above results can be used to restrict classes of $CQ^{\neg}$ (where normal forms are not required) to get a lower complexity.

Proposition 2. *The complexities can be reduced in the following case:*

- *The containment checking problem of $CQ^{\neg}$'s without built-in predicates: $Q_1 \subseteq Q_2$ can be solved in polynomial time if every positive database predicate occurs at most twice in the body of Q_1 and each negated predicate does not occur positive.*
- *The containment checking problem of $CQ^{\neg}$'s without built-in predicates: $Q_1 \subseteq Q_2$ is in NP if Q_1 is a fixed query. This keeps to be the case also for the $CQ^{\neg}$'s with built-in predicates.*

Proof. The definition of normal form of $CQ^{\neg}$ decides that if every predicate in the body of Q_1 occurs at most twice and each positive part of the negated predicate does not occur as the positive predicate, then the normal form of the query is exactly the same, that means, without inequalities, without unions of queries. As a result, it is easily seen from Saraiya's result that the containment problem of $CQ^{\neg}$ is polynomial time solvable.

When Q_1 is a fixed $CQ^{\neg}$, then the number of normal forms of Q_1 becomes polynomial: n^c where n is the maximal arity of all predicates, and c is the number of positive predicates times the number of negative predicates, which is here a constant. The containment checking now is same hard as the positive counterparts, which was proved in [KMT98] to be in NP. □

3.5 Containment of Unions of $CQ^{\neg}$'s

When the union of $CQ^{\neg}$'s is considered, the containment checking is considerably complicated. In [SY80], a theorem was given for CQ's without negation and built-in predicates: if Q is a CQ and $\mathcal{Q}$ a set of CQ's, then $Q \subseteq \mathcal{Q}$ if and only if there is a Q' in $\mathcal{Q}$ such that $Q \subseteq Q'$. However, when generalized CQ's allowing negation, the above theorem will not hold anymore.

Example 7. Consider the following queries with the EDBs *man*, *father*, and *husband*:

$$\begin{array}{lll} Q: & q(X) & :\!\!-\ man(X), \neg father(X). \\ Q_1: & q(Y) & :\!\!-\ man(Y), \neg husband(Y). \\ Q_2: & q(Z) & :\!\!-\ man(Z), \neg father(Z), husband(Z). \end{array}$$

It is easily seen, that neither $Q \subseteq Q_1$, nor $Q \subseteq Q_2$, but $Q \subseteq Q_1 \cup Q_2$. □

Because of space limitation, we provide the following lemma without proof.

Lemma 3. *Given a $CQ^{\neg}$ Q in normal form and a union of $CQ^{\neg}$'s $\mathcal{Q} = Q_1 \cup Q_2 \cup ... \cup Q_n$; if $Q \subseteq Q_1 \cup Q_2 \cup ... \cup Q_n$, then there exists a set of queries $\{Q_{n_1}, ..., Q_{n_m}\} \subseteq \mathcal{Q}$, such that $Q^+ \subseteq Q^+_{n_1} \cup ... \cup Q^+_{n_m}$.* □

$Q : q(X) : -man(X), \neg father(X).$
$Q_1 : q(Y) : -man(Y), \neg husband(Y).$ $Q_2 : q(Z) : -man(Z), \neg father(Z), husband(Z).$

Step 0

$Q' : q(X) : -man(X), husband(X), \neg father(X).$
$Q_2 : q(Z) : -man(Z), \neg father(Z), husband(Z).$

Step 1 $\{Y \rightarrow X\}$

$Q'' : q(X) : -man(X), husband(X), father(X), \neg father(X).$
Unsatisfiable

Step 2 $\{Z \rightarrow X\}$

Figure 1: The Checking Steps in Example 7

The containment test algorithm can be intuitively stated as follows: consider all the subsets of $\mathcal{Q}$ which fulfill the condition in Lemma 3 $\mathcal{S}$: for each containment mapping $\rho_1, ..., \rho_r$ from the queries in $\mathcal{S}$ to Q, the unsatisfiability of Q'' which is similar defined as in Theorem 2 will be tested. If each such Q'' is unsatisfiable, then we stop with success; if not, then the containment of from $\{\mathcal{Q} - \mathcal{S}\}$ to Q''_i must be tested recursively. The algorithm will stop in different situations: (1) when no query from the union could be selected whose positive part contains that of the query, then the algorithm will stop with failure; (2) when all such Q'''s are unsatisfiable, then the algorithm will stop with success; (3) when the union is empty and still no containment is found then the algorithm will stop with failure.

It is easy to see that the algorithm involves the exponential time of containment checking of CQ's with built-in predicates, which is already in Π_2^p. The question whether query containment of $CQ^{\neg}$'s with unions is in Π_2^p remains open.

This procedure is demonstrated w.r.t. Example 7 in Fig. 1: In the first step (Step 1), it is found that the positive part of Q is contained in that of Q_1; then the unsatisfiability of Q' is tested. Since it is still satisfiable, we continue with the query Q' and Q_2. This time, Q'^{+} is contained in Q_2^{+}, and further we have Q'' is unsatisfiable. The conclusion can be drawn that $Q \subseteq Q_1 \cup Q_2$.

4 Conclusion

We have discussed query containment for the conjunctive queries with negated subgoals $CQ^{\neg}$'s. By introducing the normal form of $CQ^{\neg}$'s, we developed a sound and complete algorithm testing the containment of $CQ^{\neg}$'s $Q_1 \subseteq Q_2$, which can be polynomially reduced to the containment checking of CQ's with built-in predicates - if Q_1 is in normal form -, which is generally Π_2^p-complete. Moreover, by giving some restriction of the predicates in queries, we reduced the complexity to polynomial by restricting the occurrence of each positive predicate to twice in Q_1 and each positive part of the negated predicate does not occur as positive predicate; to NP by taking one query as fixed. We discussed the complexity of containment checking of unions of $CQ^{\neg}$'s, and conjecture that it is higher than Π_2^p.

References

[CM77] A. K. Chandra and P. M. Merlin. Optimal implementations of conjunctive queries in relational data bases. In *Proceedings of the 9th ACM Symposium on Theory of Computing (STOC)*, pages 77–90. ACM Press, 1977.

[FLS98] D. Florescu, A. Levy, and D. Suciu. Query containment for conjunctive queries with regular expressions. In *Proceedings of the Seventeenth Symposium on Principles of Database Systems*. ACM Press, 1998.

[GSUW94] A. Gupta, Y. Sagiv, J. D. Ullman, and J. Widom. Constraint checking with partial information. In *Proceedings of the Thirteenth Symposium on Principles of Database Systems*. ACM Press, 1994.

[Klu88] A. Klug. On conjunctive queries containing inequalities. In J. ACM 35:1, pp. 146–160, 1988.

[KMT98] P. G. Kolaitis, D. L. Martin, and M. N. Thakur. On the complexity of the containment problem for conjunctive queries with built-in predicates. In *ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, 1998.

[LMSS95] A. Levy, A. O. Mendelzon, Y. Sagiv, and D. Srivastava. Answering queries using views (extended abstract). In *Proceedings of the ACM SIGACT SIGMOD SIGART Symposium on Principles of Database Systems*, volume 14, pages 95–104, 1995.

[LS93] A. Levy and Y. Sagiv. Queries independent of updates. In *19th International Conference on Very Large Data Bases*, 1993.

[LS97] A. Levy and D. Suciu. Deciding containment for queries with complex objects. In *Proceedings of PODS'97*. ACM Press, 1997.

[RSUV89] R. Ramakrishnan, Y. Sagiv, J. D. Ullman, and M. Y. Vardi. Proof-tree transformation theorems and their applications. In *Proceedings of the Eighth ACM Symposium on Principles of Database Systems*, 1989.

[Sar91] Y. Saraiya. *Subtree elimination algorithms in deductive databases*. PhD thesis, Stanford University, 1991.

[SY80] Y. Sagiv and M. Yannakakis. Equivalence among relational expressions with the union and difference operations. *Journal of the ACM, (4)*, 27(4):633–655, 1980.

[Ull89] J. Ullman. *Principles of Database and KnowledgeBase Systems, Volume II*. Computer Science Press, 1989.

[Ull97] J. D. Ullman. Information integration using logical views. In *ICDT*, 1997.

[vdM92] R. van der Meyden. The complexity of querying infinite data about linearly ordered domains. In *Proc. of the 11th ACM Symp. on PODS, pp331-345*, 1992.

[ZÖ97] X. Zhang and Z. Meral Özsoyoglu. Implication and referential constraints: A new formal reasoning. *TKDE*, 9(6):894–910, 1997.