Conjunctive Query Containment in the Presence of Disjunctive Integrity Constraints (Extended Abstract)

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

Abstract

This paper considers the problem of containment of conjunctive queries (CQ) with disjunctive integrity constraints. Query containment problem in the presence of integrity constraints has been studied broadly, especially with functional and inclusion dependencies. However, to handle incomplete information in the database, disjunctions are needed to be expressed as integrity constraints. In this paper we introduce disjunctive referential integrity constraints and give a sound and complete algorithm for checking the containment of conjunctive queries under disjunctive referential and implication constraints.

1 Introduction

The query containment problem is to check if the answer set of one query is always the subset of another query for any given database. Algorithms for query containment are crucial in several contexts in database. Starting with [CM77], who showed that the query containment of CQ is NP-complete, many researches have been working on the extension of containment algorithms for CQ's with inequalities [Klu88] [ZÖ97], unions of CQ's [SY80], and CQ's with negated subgoals [LS93], [Ull97]. Query containment on different data structures has also been considered. Complex objects are studied in [LS97], and semi-structured data with regular expression is studied in [FLS98].

Query containment problem in the presence of integrity constraints is studied first in [JK82], especially with functional and inclusion dependencies. In [ZÖ97], the integrity constraints are extended to implication constraints and referential constraints, which are the generalized form of functional constraints and inclusion dependencies, respectively. However, to handle incomplete information in the database, disjunctions are needed to be expressed as integrity constraints. The following example illustrates situation where incomplete information gives rise to disjunctive integrity constraints.

Example 1. Consider the following database:

1. The relation schemes:

   $\texttt{empl}(worker\_name), \texttt{group}(group\_name),$

$\texttt{member}(worker, group\_name, work\_name), \texttt{same\_skill}(worker, worker)$.

2. The following disjunctive constraint states that each employee belongs to one of the two groups, p_1 and p_2.

$$\textbf{drc:} \quad \texttt{empl}(X) \quad \rightarrow \quad \exists W_1\ \texttt{member}(X, p_1, W_1) \vee \exists W_2\ \texttt{member}(X, p_2, W_2).$$

3. The implication constraint states that if two workers have the same skill, then they should not belong to the same group.

$$\textbf{ic1:} \quad \texttt{same\_skill}(X, X) \rightarrow .$$
$$\textbf{ic2:} \quad \texttt{empl}(X), \texttt{empl}(Y), \texttt{same\_skill}(X, Y), \texttt{member}(X, G, W_1), \texttt{member}(Y, G, W_2) \rightarrow .$$

4. Finally, a relation q to define all pairs of employees working in different groups is composed as the union of two queries, denoted $\mathcal{Q}$, of Q_1 and Q_2:

$$Q_1: \quad q(X, Y) \quad \leftarrow \quad \texttt{empl}(X), \texttt{empl}(Y), \texttt{member}(X, p_1, W_1), \texttt{member}(Y, p_2, W_2).$$
$$Q_2: \quad q(X, Y) \quad \leftarrow \quad \texttt{empl}(X), \texttt{empl}(Y), \texttt{member}(X, p_2, W_3), \texttt{member}(Y, p_1, W_4).$$

Now we ask the query "list all the two-worker pairs who have the same skills":

$$Q: \quad q(X, Y) \quad \leftarrow \quad \texttt{empl}(X), \texttt{empl}(Y), \texttt{same\_skill}(X, Y).$$

Assuming the constraints have been enforced, that is, only relation instances satisfying **drc** and **ic1, ic2** are stored in the database, then we can get that Q is contained in the union of Q_1 and Q_2, i.e. $Q \subseteq Q_1 \cup Q_2$. From the disjunctive constraint we know, that every employee must work in $p1$ or in $p2$. The implication constraints then enforce, that employees having same skills cannot work in the same group. Therefore, any answer to our query Q is guaranteed to be an element of the union $\mathcal{Q}$. However, without the constraints enforced, this containment relationship would not hold any more. □

Moreover, in dealing with the query language like XPath for XML, disjunction is proposed in integrity constraints expressing the schema information of XML [DT01]. The next integrity constraint is taken from [DT01], stating that if a node u is the descendant of both x and y, then either x and y are the same node, or x is the descendant of y, or y is the descendant of x.

$$(\texttt{line}) \ \ \forall x, y, u \ \ [\texttt{desc}(x, u) \wedge \texttt{desc}(y, u) \rightarrow x = y \vee \texttt{desc}(x, y) \vee \texttt{desc}(y, x)]$$

In this paper we introduce disjunctive referential integrity constraints and give a sound and complete algorithm for checking the containment of conjunctive queries under disjunctive referential and implication constraints. The technique for handling disjunctive referential constraints is related to the well known minimal model semantics for disjunctive logic programming [LMR92]. Our work generalizes the results of [ZÖ97] in which only referential constraints without disjunctions are considered; we give a solution to a question left open so far. Detailed proofs of the following theorems can be found in the longer version [WL02].

2 Preliminaries

Any expression of the form $p(\bar{X})$, where p is a predicate whose vector of arguments $\bar{X}$ is built out of variables and constants, is called an *atom*. A *conjunctive query* (CQ) Q then is an expression built out of atoms in the following way:

$$q(\bar{X}) \leftarrow p_1(\bar{Y_1}), ..., p_n(\bar{Y_n})$$

where $q(\bar{X})$ is the *head*, and $p_1(\bar{Y_1}), ..., p_n(\bar{Y_n})$ is the *body* of the query. We assume that the variables appearing in the head also appear in the body. The queries in Example 1 are CQs. A CQ is *applied* to the database D (written as $Q(D)$) by considering all possible substitutions of values for the variables in the body. If a substitution makes all the subgoals true, then the same substitution, applied to the head, is the element of the *answer set*, which contains tuples. A conjunctive query Q_1 is contained in another conjunctive query Q_2, denoted as $Q_1 \subseteq Q_2$, if for all databases D, $Q_1(D) \subseteq Q_2(D)$. Two CQs are equivalent if and only if each is contained in the other.

If inequalities (or built-in predicates) are allowed in a conjunctive query, the form of CQ is extended as follows:

$$q(\bar{X}) \leftarrow p_1(\bar{Y_1}), ..., p_n(\bar{Y_n}), I.$$

where I is the conjunct of formulas of the form $(u_1 \; op \; u_2)$, in which both u_1 and u_2 can be constants or variables, and if any u is a variable, then u is in $\{\bar{Y_1}, ..., \bar{Y_n}\}$. The containment definition of CQs with inequalities is the same as that of CQs.

Definition 1 (Implication Constraint). *[ZÖ97] An implication constraint is a formula of the form*

$$p_1(\bar{Y_1}), ..., p_n(\bar{Y_n}), I \rightarrow .$$

where $p_1(\bar{Y_1}), ..., p_n(\bar{Y_n})$ are atoms and I is a conjunction of inequalities as defined above. Note that implication constraints are called denial constraints too. □

Definition 2 (Disjunctive Referential Constraint). *A disjunctive referential constraint (DRC) is an expression of the form*

$$\forall (Y_1, ..., Y_m) \quad [r(Y_1, ..., Y_m) \quad \rightarrow \quad \exists (\bar{Z_1}) s_1(\bar{X_1}) \vee ... \vee \exists (\bar{Z_u}) s_u(\bar{X_u})].$$

where $s_1, ..., s_u (1 \leq u)$, and r are predicate names; $Y_1, ..., Y_m$ and $\bar{Z_1}, ..., \bar{Z_u}$ are different variables. $\bar{X_1}, ..., \bar{X_u}$ are tuples of variables and constants; for any variable $V \in \bar{X_i} (1 \leq i \leq u)$, if $V \notin \{Y_1, ..., Y_m\}$, then $V \in \bar{Z_i} (1 \leq i \leq u)$. Note that if $u = 1$, the constraint will be reduced to a referential constraint as described in [ZÖ97]. □

In first order logic, skolemization is used to eliminate existential quantifications without loss of information. For example, the skolemized form of the constraint the **drc** in Example 1 is:

$$\texttt{empl}(X) \quad \rightarrow \quad \texttt{member}(X, p1, f_1(X)) \vee \texttt{member}(X, p2, f_2(X)).$$

where f_1 and f_2 are unique Skolem function symbols.

When clear from the context, the set of disjunctive referential constraints is denoted by $\mathcal{DRC}$ and the set of implication constraints is denoted by $\mathcal{IC}$. Given a database D, we define that if D is consistent with $\mathcal{DRC}$ and $\mathcal{IC}$ if D *satisfies* the integrity constraints. Next we give the formal definition of *consistency*.

Definition 3 (Consistency). *A database instance D is consistent if D satisfies $\mathcal{DRC}$ and $\mathcal{IC}$ in the standard model-theoretic sense, that is, $D \models \{\mathcal{DRC}, \mathcal{IC}\}$; D is inconsistent otherwise.* □

Definition 4. *A conjunctive query Q is $\{\mathcal{DRC}, \mathcal{IC}\}$ – contained in another conjunctive query Q', denoted $Q \subseteq_{\mathcal{DRC},\mathcal{IC}} Q'$, if $Q(D) \subseteq Q'(D)$ for any database D consistent with the integrity constraints $\mathcal{DRC}, \mathcal{IC}$. Q and Q' are $\{\mathcal{DRC}, \mathcal{IC}\}$ – equivalent, denoted $Q =_{\mathcal{DRC},\mathcal{IC}} Q'$ if $Q \subseteq_{\mathcal{DRC},\mathcal{IC}} Q'$ and $Q \supseteq_{\mathcal{DRC},\mathcal{IC}} Q'$.* □

3 Query Containment with $\mathcal{DRC}$ and $\mathcal{IC}$

In [ZÖ97], a *referential expansion* is introduced to rewrite the original query to a unique expanded query, which reflects a respective referential constraint. However, in the case of a disjunctive referential constraint, we obtain a set of expanded queries. The technique we shall apply borrows from the minimal model semantics for disjunctive logic programming [LMR92].

3.1 Disjunctive Referential Expansion

Definition 5 (Disjunctive Referential Expansion). *Let $\mathcal{DRC}$ be a set of disjunctive referential constraints and Q a conjunctive query of the form*

$$q(\bar{X}) \quad \leftarrow \quad p_1(\bar{Y_1}), ..., p_n(\bar{Y_n}), I.$$

Let F denote the set of atoms in the body of Q, namely, $\{p_1(\bar{Y_1}), ..., p_n(\bar{Y_n})\}$, and I a conjunction of inequalities.

1. *Let M be any set of atoms such that for any atom T of M, if there is a DRC rule in $\mathcal{DRC}$ of the form as in Definition 2 and a substitution ρ from $r(Y_1, ..., Y_m)$ to T, then there is at least one of $\rho(s_i(\bar{X_i}))(1 \leq i \leq u)$ in M.*
2. *Let $\mathcal{M}$ be the set of all such M, for which in addition there holds $F \subseteq M$.*
3. *Let $min(\mathcal{M}) = \{M \in \mathcal{M} : \nexists M' \in \mathcal{M}, M' \subset M\}$.*
4. *We enumerate the elements of $min(\mathcal{M})$ by $\{F'_1, \ldots, F'_k\}$.*
5. *We give each skolem function in $F'_i(1 \leq i \leq k)$ to a distinct variable name which does not occur in $F'_i(1 \leq i \leq k)$. Finally the set $\{F_1, \ldots, F_k\}$ is obtained by the renaming of each element in $\{F'_1, \ldots, F'_k\}$.*

The Disjunctive Referential Expansion of Q using $\mathcal{DRC}$ is the set of sub-queries denoted $\mathcal{Q}^e = (Q^e_1, Q^e_2, ..., Q^e_k)$. Each Q^e_i has the form $q(\bar{X}) \leftarrow F_i, I.$, where F_i is the set of atoms as defined above. □

Note that at the last step, we simply replace each skolem function with a distinct variable, so that the expanded sub-queries fall into the category of function free conjunctive queries. We argue that since such new generated variables do not appear at the head of each expanded query, using distinct variables is a natural way expressing the existential quantifiers in the $\mathcal{DRC}$. The semantical correctness is proved in [AHV95].

It should be noticed that if there are more than one DRC rules, then the *disjunctive referential expansion* is not trivial any more. Example 2 shows one expansion.

Example 2. Let $\texttt{teach}(X, Y)$ be the relational schema meaning someone X teaches the course Y, and $\texttt{emp}(X)$ that X is a employee, and so on. There are two disjunctive referential constraints as follows:

$$\begin{array}{lll} \mathbf{drc1}: & \texttt{teach}(X,Y) & \rightarrow \texttt{graduate}(X) \vee \texttt{faculty}(X). \\ \mathbf{drc2}: & \texttt{emp}(X) & \rightarrow \texttt{faculty}(X) \vee \texttt{staff}(X). \end{array}$$

The constraint **drc1** can be explained as: if someone teaches one course, then he must be either a graduate or a faculty. The constraint **drc2** means that if an employee must be either a faculty or a staff. Now the query is given of getting the people who teach a course and is also an employee of the university:

$Q : q(X) \leftarrow \texttt{teach}(X, Y), \texttt{emp}(X).$

If we *expand* the body of the query Q: , using the constraints **drc1** and **drc2** above, then the final expansion set consists of two sub-queries:

$\mathcal{Q}^e = \{Q^e_1, Q^e_2\}$, where

$$\begin{array}{llll} Q^e_1: & q(X) & \leftarrow & \texttt{teach}(X,Y), \texttt{emp}(X), \texttt{faculty}(X). \\ Q^e_2: & q(X) & \leftarrow & \texttt{teach}(X,Y), \texttt{emp}(X), \texttt{graduate}(X), \texttt{staff}(X). \end{array}$$

Note that since the atom $\texttt{faculty}(X)$ appears in both disjunctive referential constraints, there is only one sub-query that contains it, from the definition of *minimal*. □

From the above definition and example, it is easily seen that the general algorithm of *model generation*, which collects all the minimal models of a disjunctive logic programming can be used for the expansion here. The only difference is that the elements in F are not ground atoms. However, this can be circumnavigated by treating all the variables in F as distinct constants.

Theorem 1. *The disjunctive referential expansion can be polynomially reduced to the problem of getting the all minimal models in disjunctive logic programming (DLP).* □

Termination. The general referential expansion procedure does not terminate [ZÖ97, AHV95]. This is the case of disjunctive referential expansion too, since the referential constraints are the special form of the disjunctive referential constraints.
However, if the $\mathcal{DRC}$ has the *acyclic* property, the expansion will always terminate.

Definition 6. *A set of* $\mathcal{DRC}$ *is* acyclic *if there is no such sequence* $r_i(\bar{Y}_i) \rightarrow S_{1,i}(\bar{X}_{1,i}) \vee \ldots \vee S_{u_i,i}(\bar{X}_{u_i,i}) (i \in [1,n])$ *in* $\mathcal{DRC}$ *that for* $i \in [1,n], S_{l,i} = r_{i+1}$ *for* $i \in [1, n-1], l \in [1, u_i]$*, and* $S_{l,n} = r_1 (l \in [1, u_n])$*.* □

Proposition 1. *Given a conjunctive query* Q *and a set of* acyclic $\mathcal{DRC}$*, the expansion of* Q *using* $\mathcal{DRC}$ *terminates after an exponentially bounded number of steps.* □

A DRC is *full* constraint if it has no existential quantifiers. The disjunctive expansion using a set of *full* $\mathcal{DRC}$ terminates – the constraints need not to be *acyclic*. This is because the reduced form has the semantics of Disjunctive Datalog which guarantees termination [FM92, Min92].

Corollary 1. *Let* Q *be a conjunctive query and* $\mathcal{DRC}$ *a set of disjunctive referential constraints, and* $\mathcal{Q}^e = (Q_1^e, Q_2^e, ..., Q_k^e)$ *is the union of disjunctive referential expansion; given any database* D *consistent with the integrity constraints* $\mathcal{DRC}$*, if one tuple* $t \in Q(D)$*, then there is at least one* $Q_i^e \in \mathcal{Q}^e (1 \leq i \leq k)$*, such that* $t \in Q_i^e(D)$*.* □

Theorem 2. *Given a query* Q*, a set of queries* $\mathcal{V}$ *with the same format as* Q*, a set of* $\mathcal{DRC}$*;* $\mathcal{Q}^e = (Q_1^e, Q_2^e, ...Q_k^e)$ *is the union of the disjunctive referential expansion of* Q*, then* $Q \subseteq \mathcal{V}$ *in the presence of* $\mathcal{DRC}$ *(written as* $Q \subseteq_{\mathcal{DRC}} \mathcal{V}$*), if and only if for each* $Q_i^e (1 \leq i \leq k)$*, there is* $Q_i^e \subseteq \mathcal{V}$*.* □

3.2 Containment Checking Algorithm

Considering the presence of both constraints, the containment checking is processed in two steps: (i) Firstly we expand the query to an equivalent set of sub-queries using the disjunctive expansion; (ii) Secondly the containment checking under implication constraints of each sub-query is executed. It is formalized as follows:

Theorem 3. *Given a query* Q*, a set of queries* $\mathcal{V}$ *with the same format as* Q*, a set of* $\mathcal{DRC}$ *and a set of* $\mathcal{IC}$*;* $\mathcal{Q}^e = (Q_1^e, Q_2^e, ...Q_k^e)$ *is the union of the disjunctive referential expansion of* Q*.* $Q \subseteq_{\mathcal{DRC},\mathcal{IC}} \mathcal{V}$ *if and only if for each* $Q_i^e, (1 \leq i \leq k)$*, there is* $Q_i^e \subseteq_{\mathcal{IC}} \mathcal{V}$ □

According to [ZÖ97], $Q_i^e \subseteq_{\mathcal{IC}} \mathcal{V}$ means $Q_i^e \subseteq \mathcal{V}$ in the presence of $\mathcal{IC}$. To test, there must be *symbol mappings* from the $\mathcal{V}$ or $\mathcal{IC}$ to Q_i^e. For the case of inequalities in the query, there must be an implication test from the inequalities of Q_i^e to the disjunction of that of $\mathcal{V}$ [ZÖ97].

The next example illustrates the containment checking algorithm in the presence of both disjunctive referential and implication constraints:

Example 3. Given the **drc** and query Q as in Example 1, the disjunctive expansion of Q is the union of the four sub-queries:

$$\begin{array}{lll} Q_1^e: \; q(X,Y) & \leftarrow & \texttt{empl}(X), \texttt{empl}(Y), \texttt{same\_skill}(X,Y), \\ & & \texttt{member}(X,p_1,X_1), \texttt{member}(Y,p_1,Y_1). \\ Q_2^e: \; q(X,Y) & \leftarrow & \texttt{empl}(X), \texttt{empl}(Y), \texttt{same\_skill}(X,Y), \\ & & \texttt{member}(X,p_1,X_1), \texttt{member}(Y,p_2,Y_2). \\ Q_3^e: \; q(X,Y) & \leftarrow & \texttt{empl}(X), \texttt{empl}(Y), \texttt{same\_skill}(X,Y), \\ & & \texttt{member}(X,p_2,X_2), \texttt{member}(Y,p_1,Y_1). \\ Q_4^e: \; q(X,Y) & \leftarrow & \texttt{empl}(X), \texttt{empl}(Y), \texttt{same\_skill}(X,Y), \\ & & \texttt{member}(X,p_2,X_2), \texttt{member}(Y,p_2,Y_2). \end{array}$$

Note that we replace $f_1(X)$ with X_1, $f_2(X)$ with X_2, $f_1(Y)$ with Y_1 and $f_2(Y)$ with Y_2 respectively.

The containment tests will give the following results: There is a containment mapping from **ic** to Q_1^e and Q_4^e; $Q_2^e \subseteq Q_1$; $Q_3^e \subseteq Q_2$. As a result, we get that $Q \subseteq_{\mathcal{DRC},\mathcal{IC}} Q_1 \cup Q_2$. □

3.2.1 Complexity

The expansion of a conjunctive query Q using a set of *acyclic* $\mathcal{DRC}$ is decidable, but intractable, since the implication of an inclusion dependency (ind) by an acyclic set of ind's is NP-complete [AHV95]. In dealing with the *full* disjunctive expansion, the complexity of expansion is equivalent to that of minimal model generation of Disjunctive Datalog, which has been proved to be $\prod_2^p$-complete[DEGV97], however, the *data complexity* here is in terms of the size of the query. Usually the size of the query is very small compared to that of database. Therefore, our techniques are still of practical interest.

4 Conclusion

Disjunctive integrity constraints are crucial dealing with incomplete information in the database [BLR00]. Actually, it is the general form of integrity constraints introduced in [GGGM98]. Referring to the minimal model semantics of DLP, we were able to solve the query containment problem under disjunctive referential and implication constraints. As mentioned in the introduction, in dealing with semi-structured data and XML, the referential constraints in the form of disjunction have been proposed in the work of the translation of XPath to relational query model [DT01]. An extension of the *chase* algorithm is given in [DT01]. However, we argue that without using the *minimal* semantics, the expanded chase tree could have an exponential blow-up in the size of disjunctive referential constraints. Furthermore, the referential and implication constraints are the generalized form of inclusion and functional dependencies, so that our method can deal with the problem in [DT01], but not vice versa.

References

[AHV95] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases.* Addison-Wesley Publishing Company, Reading, Massachusetts, 1995.

[BLR00] Francesco Buccafurri, Nicola Leone, and Pasquale Rullo. Enhancing disjunctive datalog by constraints. *Knowledge and Data Engineering*, 12(5):845–860, 2000.

[CM77] A. K. Chandra and P. M. Merlin. Optimal implementations of conjunctive queries in relational data bases. In *STOC*, ACM Press, 1977.

[DEGV97] Evgeny Dantsin, Thomas Eiter, Georg Gottlob, and Andrei Voronkov. Complexity and expressive power of logic programming. In *IEEE Conference on Computational Complexity*, pages 82–101, 1997.

[DT01] A. Deutsch and V. Tannen. Containment and Integrity Constraints for XPath Fragments. In *KRDB*, 2001.

[FLS98] Daniela Florescu, Alon Y. Levy, and Dan Suciu. Query containment for conjunctive queries with regular expressions. In *PODS'98*. ACM Press, 1998.

[FM92] Jose Alberto Fernandez and Jack Minker. Disjunctive deductive databases. In *Logic Programming and Automated Reasoning*, pages 332–356, 1992.

[GGGM98] P. Godfrey, J. Grant, J. Gryz, and J. Minker. Integrity constraints: Semantics and applications. In *Logics for Databases and Information Systems.* Kluwer, 1998.

[JK82] David S. Johnson and Anthony C. Klug. Testing containment of conjunctive queries under functional and inclusion dependencies. In *PODS'82.* ACM, 1982.

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

[LMR92] J. Lobo, J. Minker, and A. Rajasekar. *Foundations of Disjunctive Logic Programming.* MIT press, Cambridge, Massachusetts, 1992.

[LS93] Alon Y. Levy and Yehoshua Sagiv. Queries independent of updates. In Rakesh Agrawal, Seán Baker, and David A. Bell, editors, *VLDB*, 1993.

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

[Min92] Jack Minker. Semantics of disjunctive deductive databases. In *1992 Internat. Conference on Database Theory, Berlin*, 1992.

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

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

[WL02] F. Wei and G. Lausen. Conjunctive query containment in the presence of disjunctive integrity constraints. Technical Report, University of Freiburg, 2002. *http://www.informatik.uni-freiburg.de/~dbis/Publications/02/tech_167.ps.gz.*

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