To appear in the Proceedings of the Fourteenth National Conference on Artificial Intelligence, AAAI-97. Providence, RI, 1997.

Query Optimization Using Local Completeness

Oliver M. Duschka
Department of Computer Science
Stanford University
Stanford, CA 94305
duschka@cs.stanford.edu

Abstract

We consider the problem of query plan optimization in information brokers. Information brokers are programs that facilitate access to collections of information sources by hiding source-specific peculiarities and presenting uniform query interfaces. It is unrealistic to assume that data stored by information sources is complete. Therefore, current implementations of information brokers query all possibly relevant information sources in order not to miss any answers. This approach is very costly. We show how a weaker form of completeness, local completeness, can be used to minimize the number of accesses to information sources.

Introduction

We consider the problem of query plan optimization in information integration. The goal of information integration is to provide the illusion that data stored by distributed information sources is stored in a single "global" database. Users can pose queries in terms of the global database scheme. These queries then need to be translated into queries that can be answered by the information sources. There are basically two approaches to information integration. Either the relations of the global scheme are defined in terms of the relations stored by the information sources (query-centric approach), or the relations stored by the information sources are described in terms of the global scheme (source-centric approach).

The TSIMMIS project (Chawathe *et al.* 1994) investigates the query-centric approach to information integration. Query planning is very efficient using this approach, because user queries simply have to be matched against query templates to find the corresponding predefined query plans. However, the query-centric approach has two major disadvantages. The number of possible user queries is restricted, and adding new information sources requires rewriting all related query templates.

The Information Manifold (Kirk *et al.* 1995) and the Infomaster project (Geddis *et al.* 1995) follow the more flexible source-centric approach. This approach is very well suited for dynamic environments like the Internet, because adding, removing, or changing an information source only requires adding, removing, or changing the description of this respective information source. Because query plans have to be computed at query time, efficient query plan generation and optimization become crucial.

While query plan generation in the source-centric approach has been studied extensively (Levy *et al.* 1995; Rajaraman, Sagiv, & Ullman 1995; Levy, Rajaraman, & Ordille 1996a; 1996b; Duschka & Genesereth 1997a; 1997b; Duschka & Levy 1997), little work has been done on query plan optimization. We show how *local completeness* information as introduced in (Etzioni, Golden, & Weld 1994) and explained in the following can be used for query plan optimization in the source-centric approach to information integration.

Local completeness

Information brokers communicate with users in terms of a global scheme consisting of a set of *world-relations* $p_1, p_2, \ldots, p_m$. The broker has access to a number of information sources. We refer to these sources as $IS_1, \ldots, IS_n$. Each information source IS_i is assumed to store a *site-relation* s_i. Ideally, a site-relation would be a materialized view defined in terms of world-relations. This view then would concisely describe the data stored by the information source. However, this requirement is seldom satisfiable in real world applications.

Example 1 Assume an information broker wants to integrate classified ads from several newspapers. One of the world-relations then might be a relation

$$cars\_for\_sale(Manufacturer, Model, Year, Mileage, Price, Phone\_number)$$

representing information found in used car classifieds. Because a specific used car classified can appear in any of the newspapers — the newspapers have overlapping markets — no newspaper is "complete" on some part of the *cars_for_sale* relation. The best one can do in describing the information sources is to state that the data they store is contained in the *cars_for_sale* relation. □

Because frequently site-relations do not correspond to materialized views in terms of the world-relations, implementations of information brokers (Kirk *et al.* 1995; Geddis *et al.* 1995) consider the data stored by an information source to be *contained* in the corresponding view. Using this interpretation, however, an information broker might be forced to retrieve much redundant information. If several information sources store data that might be relevant to a user query, then all of these sources need to be queried, although data stored by one information source might be completely stored by another.

In some applications it is impossible to improve on this situation. If for example a user asks for used red sports cars, then there is no way to tell which newspaper might provide matching classifieds. In many application domains, however, it is known that some subset of the data that an information source stores is complete, although the entire data stored by the information source might not be complete. This so called *local completeness information* can be used to minimize the number of information sources that need to be queried. We represent an information source therefore by *two* views: a *conservative view* v_i^c as a lower bound of s_i, and a *liberal view* v_i^l as an upper bound of s_i. Conservative views describe the subsets of the data that are known to be complete, i.e. they encode local completeness information.

Conservative and liberal views have the same scheme as the corresponding site-relations. If t is a tuple belonging to the conservative view, then t is indeed stored by the information source. If t is a tuple stored by the information source, then t also belongs to the liberal view. In the special case that an information source indeed stores a materialized view in terms of world-relations, the corresponding conservative and liberal views are identical.

Example 2 Assume an information broker wants to integrate sources that provide information on the current market value of cars. The information broker might export a world-relation like

$$bluebook(Manufacturer, Model, Year, Value).$$

Assume an infomation source stores current market values for cars, and guarantees that it has all information for models built after 1990. This information source can be descibed as follows:

$$v^c_{info}(Ma, Mo, Ye, Va) :- bluebook(Ma, Mo, Ye, Va),\ Ye > 1990$$

$$v^l_{info}(Ma, Mo, Ye, Va) :- bluebook(Ma, Mo, Ye, Va)$$

A second information source accessible by the information broker might be a database of the car manufacturer BMW. This database stores information on all BMW models, and nothing else. The completeness of this database can be expressed by coinciding conservative and liberal views:

$$v^{c,l}_{bmw}(bmw, Mo, Ye, Va) :- bluebook(bmw, Mo, Ye, Va)$$

If a user requests information on a car build after 1990 or built by BMW, then only information source IS_{info} or IS_{bmw} respectively needs to be queried. On the other hand, if a user asks for all cars with market value over $\$50,000$, then both information sources have to be queried in order not to miss any answers. □

Retrievable queries

Given a user query, an information broker needs to figure out which information sources to retrieve the requested data from. The available information sources $IS_1, \ldots, IS_n$ store site-relations $s_1, \ldots, s_n$ that are described using conservative views $v_1^c, \ldots, v_n^c$ and liberal views $v_1^l, \ldots, v_n^l$. Using these descriptions, the information broker needs to translate the user query into a query that can be answered by querying the information sources and possibly internally post-processing the results. Let us call a query that only involves site-relations and built-in predicates a *retrievable query*. In the following, we are going to define three important properties of retrievable queries: semantical correctness, source-completeness, and view-minimality. The algorithms presented in (Qian 1996; Levy, Rajaraman, & Ordille 1996a; 1996b; Duschka & Genesereth 1997a; 1997b) generate semantically correct and source-complete retrievable queries. We will show how these algorithms can be extended to also guarantee view-minimality.

Semantical correctness

The most basic requirement a retrievable query r must satisfy in order to qualify as an answer to a user query q is that every tuple reported to the user by executing r does satisfy q. A retrievable query r is *semantically correct* with respect to a user query q, if r is contained

in q for all instances of the site-relations $s_1, \ldots, s_n$ consistent with the given conservative and liberal views.

Example 3 Assume a user asks for used cars built in 1991 that are offered for sale below their current market value:

$$q(Ma, Mo, Mi, Pr, Ph) :- \\ cars\_for\_sale(Ma, Mo, 1991, Mi, Pr, Ph), \\ bluebook(Ma, Mo, 1991, Va),\ Pr < Va$$

In addition to the two information sources in example 2, the information broker might have access to the used car classifieds of the San Francisco Chronicle and the San Jose Mercury News. These two information sources don't guarantee any local completeness, and are therefore only described by the following liberal views:

$$v^l_{sfc}(Ma, Mo, Ye, Mi, Pr, Ph) :- \\ cars\_for\_sale(Ma, Mo, Ye, Mi, Pr, Ph)$$

$$v^l_{sjmn}(Ma, Mo, Ye, Mi, Pr, Ph) :- \\ cars\_for\_sale(Ma, Mo, Ye, Mi, Pr, Ph)$$

The retrievable query

$$r_1(Ma, Mo, Mi, Pr, Ph) :- \\ s_{sfc}(Ma, Mo, 1991, Mi, Pr, Ph), \\ s_{bmw}(Ma, Mo, 1991, Va),\ Pr < Va$$

is semantically correct with respect to q. It is essential to add the selection on year and price range in order for the query to be semantically correct. We assume that information sources have the capability of equality selection. Therefore, the selection of used cars built in 1991 can be pushed to the sources. However, the selection on the price range needs to be added as a post-processing step in the information broker. □

Source-completeness

A user will hardly be satisfied by an answer from an information broker that is guaranteed merely to be semantically correct. For example, answering with the empty set is always semantically correct. Indeed, users require that they obtain all information from the broker that they could get by manually checking the sources. The notion of source-completeness formalizes this demand for a "best possible" retrievable query. A retrievable query r is *source-complete* if every semantically correct retrievable query r' is contained in r for all instances of the site-relations $s_1, \ldots, s_n$ consistent with the given conservative and liberal views.

Example 4 The retrievable query in example 4 is *not* source-complete. A used BMW offered for sale in the San Jose Mercury News, for example, will not be contained in the answer of r_1 although it might be in the answer of the retrievable query

$$r_2(Ma, Mo, Mi, Pr, Ph) :- \\ s_{sjmn}(Ma, Mo, 1991, Mi, Pr, Ph), \\ s_{bmw}(Ma, Mo, 1991, Va),\ Pr < Va.$$

The union of the retrievable queries r_1 and r_2 is still not source-complete, because all cars in the answer are manufactured by BMW. Information source IS_{info} can be used to also consider cars of other manufacturers:

$$r_3(Ma, Mo, Mi, Pr, Ph) :- \\ s_{sfc}(Ma, Mo, 1991, Mi, Pr, Ph), \\ s_{info}(Ma, Mo, 1991, Va),\ Pr < Va.$$

$$r_4(Ma, Mo, Mi, Pr, Ph) :- \\ s_{sfmn}(Ma, Mo, 1991, Mi, Pr, Ph), \\ s_{info}(Ma, Mo, 1991, Va),\ Pr < Va.$$

The retrievable query that returns the union of r_1, r_2, r_3, and r_4 is indeed source-complete. □

View-minimality

By just executing *all* semantically correct retrievable queries, assuming there is only a finite number, and reporting the union of all answers to the user, the given answer would be guaranteed to be source-complete. On the other hand, much information might be retrieved redundantly from several information sources. A retrievable query requiring considerably fewer information sources might still be source-complete. A retrievable query r is *view-minimal* if every semantically correct and source-complete retrievable query r' queries at least as many information sources as r.

Example 5 Because IS_{info} is guaranteed to store all information for cars built after 1990, there is no information in the BMW database for cars built in 1991 that could not be found in IS_{info}. Therefore, retrievable queries r_1 and r_2 are redundant. The retrievable query that is semantically correct, source-complete, and view-minimal is the union of r_3 and r_4. □

Computing with source descriptions

Preliminaries

A *conjunctive query* is an expression of the form

$$q(\bar{X}) :- p_1(\bar{X}_1), \ldots, p_n(\bar{X}_n),$$

where $p_1, \ldots, p_n$ are relation names, and $\bar{X}, \bar{X}_1, \ldots, \bar{X}_n$ are tuples of variables and constants such that any variable appearing in $\bar{X}$ appears also in $\bar{X}_1, \ldots, \bar{X}_n$.

In this paper, liberal views are conjunctive queries. Conservative views and user queries can be unions of conjunctive queries. Moreover, conservative views and user queries can contain built-in order predicates like "$<$", "$=$", and "$\neq$".

Let us denote by $q(D)$ the result of evaluating query q on database D. Given two queries q_1 and q_2, we say that q_1 is *contained* in q_2 if for every database D,

$q_1(D)$ is contained in $q_2(D)$. Algorithms for testing containment of conjunctive queries, unions of conjunctive queries, and unions of conjunctive queries with built-in order predicates are described in (Chandra & Merlin 1977), (Sagiv & Yannakakis 1980), and (Klug 1988) respectively.

Need for syntactic criteria

Given conservative and liberal views $v_1^c, \ldots, v_n^c$, $v_1^l, \ldots, v_n^l$ and a user query q, the goal is to generate a query r that satisfies the following four conditions:

- *(retrievability)* r uses only $s_1, \ldots, s_n$ and built-in predicates.
- *(semantical correctness)* For every database D_w over the world-relations and every database D_s over the site-relations with $v^c(D_w) \subseteq D_s \subseteq v^l(D_w)$, $r(D_s)$ is contained in $q(D_w)$.
- *(source-completeness)* For every database D_w over the world-relations and every database D_s over the site-relations with $v^c(D_w) \subseteq D_s \subseteq v^l(D_w)$, $r'(D_s)$ is contained in $r(D_s)$ for every semantically correct retrievable query r'.
- *(view-minimality)* $|r| \leq |r''|$ for every semantically correct and source-complete retrievable query r''.

Here $|r|$ denotes the number of information sources required in the retrievable query r.

The test of both semantical correctness and source-completeness is relative to databases D_s over the site-relations with the restriction that $v^c(D_w) \subseteq D_s \subseteq v^l(D_w)$ The only way to effectively test semantical correctness and source-completeness is to use the descriptions of the site-relations given by the conservative and liberal views. We therefore have to develop criteria for semantical correctness and source-completeness that do not refer to a restricted set of databases over the site-relations.

Syntactic criterion for semantical correctness

In order to test semantical correctness it is necessary to test containment of a retrievable query in a user query. Retrievable queries are formulated in terms of site-relations. User queries, on the other hand, are formulated in terms of world-relations. In order to compare queries in different languages, we have to translate one into the language of the other. Let us denote a retrievable query r requiring site-relations $s_{i_1}, \ldots, s_{i_k}$ as $r[s_{i_1}, \ldots, s_{i_k}]$. Replacing each occurrence of s_{i_j} by the corresponding body of the definition of $v_{i_j}^l$ yields $r[v_{i_1}^l, \ldots, v_{i_k}^l]$ which we denote as $r[s \mapsto v^l]$.

Example 6 If r is the retrievable query

$$r(X,Y) :- s_1(X,Z),\ s_2(Z,Y,c),\ X < 100$$

and the liberal views corresponding to site-relations s_1 and s_2 are

$$v_1^l(X,Y) :- p_1(X,Y,Z,d) \quad \text{and}$$
$$v_2^l(X,Y,Z) :- p_2(X,Y),\ p_3(Y,Z),$$

then $r[s \mapsto v^l]$ denotes the query

$$r[s \mapsto v^l](X,Y) :- p_1(X,Z,Z',d),\ p_2(Z,Y),\ p_3(Y,c),\ X < 100.$$

□

Semantical correctness requires that the answer of a retrievable query is guaranteed to satisfy the given user query. Because the site-relations themselves are unknown to the information broker, they must be assumed to be possibly as large as indicated by the liberal views. This intuition motivates the following syntactic characterization of semantical correctness:

Theorem 1 *A retrievable query r is semantically correct with respect to q if and only if $r[s \mapsto v^l]$ is contained in q.*

Proof. If r is semantically correct with respect to q, then for every database D_w over the world-relations, $r(D_s) \subseteq q(D_w)$ for the database $D_s = v^l(D_w)$ over the site-relations, and therefore $r[s \mapsto v^l] \subseteq q$. If r is not semantically correct with respect to q, then there is a database D_w over the world-relations, a database D_s over the site-relations with $v^c(D_w) \subseteq D_s \subseteq v^l(D_w)$, and a tuple t in $r(D_s)$ that is not in $q(D_w)$. Since unions of conjunctive queries are monotone, adding tuples to D_s until $D_s = v^l(D_w)$ will not delete t from $r(D_s)$. Therefore, $r[s \mapsto v^l]$ is not contained in q. □

Syntactic criterion for source-completeness

Intuitively, a retrievable query r is source-complete if all information asked for by a user query and available from site-relations is retrieved. No other semantically correct retrievable query should be able to retrieve more information than r. We were able to formulate a syntactic criterion for semantical correctness by replacing all occurrences of site-relations by their liberal descriptions. One might hope to find a similar criterion for source-completeness. If $r'[s \mapsto s^l]$ is contained in $r[s \mapsto v^c]$, then r' does not need to be retrieved, because all information that might possibly be retrieved using r' is guaranteed to be retrieved using r. This observation suggests that source-completeness of r might be equivalent to the condition "$r'[s \mapsto v^l]$ is contained in $r[s \mapsto v^c]$ for all semantically correct retrievable queries r'". This condition is sufficient for source-completeness. It is not necessary, however, as can be seen from the following example.

Example 7 Assume there are three site-relations described by the following conservative and liberal views:

$$v_1^c(X) :- p_1(X),\ p_2(X,a)$$
$$v_1^l(X) :- p_1(X),\ p_2(X,Y)$$
$$v_2^c(X) :- p_1(X),\ p_2(X,Y)$$
$$v_2^l(X) :- p_1(X)$$
$$v_3^c(X) :- p_3(X,a)$$
$$v_3^l(X) :- p_3(X,Y)$$

Given the user query

$$q(X) :- p_1(X),\ p_3(X,Z)$$

the two retrievable queries

$$r_1(X) :- s_1(X),\ s_3(X) \quad \text{and}$$
$$r_2(X) :- s_2(X),\ s_3(X)$$

are semantically correct with respect to q. Retrievable query r_2 is source-complete, because site-relation s_1 is guaranteed to be contained in site-relation s_2. However,

$$r_1[s \mapsto v^l](X) :- p_1(X),\ p_2(X,Y),\ p_3(X,Z)$$

is not contained in

$$r_2[s \mapsto v^c](X) :- p_1(X),\ p_2(X,Y),\ p_3(X,a).$$

□

The problem in example 7 is that site-relation s_3 appears both in r_1 and in r_2. Although s_3 is of course contained in s_3, v_3^l is not contained in v_3^c. A small variation on this idea, however, provides us with a syntactic criterion for source-completeness. If r is a retrievable query, then let $r[s \mapsto s \wedge v^l]$ be the result of replacing every site-relation s_i in r with the conjunction of s_i and the corresponding body of the definition of v_i^l. Let $r[s \mapsto s \vee v^c]$ be the result of replacing every site-relation s_i in r with the disjunction of s_i and the corresponding body of the definition of v_i^c.

Example 8 Continuing with example 7, $r_1[s \mapsto s \wedge v^l]$ denotes the query

$$r_1[s \mapsto s \wedge v^l](X) :- s_1(X),\ s_3(X),\ p_1(X)$$
$$p_2(X,Y),\ p_3(X,Z)$$

and $r_2[s \mapsto s \vee v^c]$ denotes the query $r_{21} \cup r_{22} \cup r_{23} \cup r_{24}$ with

$$r_{21}(X) :- s_2(X),\ s_3(X)$$
$$r_{22}(X) :- s_2(X),\ p_3(X,a)$$
$$r_{23}(X) :- p_1(X),\ p_2(X,Y),\ s_3(X)$$
$$r_{24}(X) :- p_1(X),\ p_2(X,Y),\ p_3(X,a)$$

Because $r_1[s \mapsto s \wedge v^l]$ is contained in r_{23}, $r_1[s \mapsto s \wedge v^l]$ is contained in $r_2[s \mapsto s \vee v^c]$. As we will show in the following theorem, this implies that r_1 is redundant. □

Although both $r[s \mapsto s \wedge v^l]$ and $r[s \mapsto s \vee v^c]$ still contain site-relations, we can use these two notions for a syntactic criterion for source-completeness. The reason is that database D_s is no longer constrained to satisfy $v^c(D_w) \subseteq D_s \subseteq v_i^l(D_w)$, but can be chosen arbitrarily. This means that each site-relation can be treated as just another world-relation, and containment can be tested without referring to a restricted set of databases over the site-relations.

Theorem 2 *A retrievable query r is source-complete if and only if for every retrievable query r' with $r'[s \mapsto v^l] \subseteq q$, $r'[s \mapsto s \wedge v^l]$ is contained in $r[s \mapsto s \vee v^c]$.*

Proof. Let r be a source-complete retrievable query, and assume r' is a retrievable query with $r'[s \mapsto v^l] \subseteq q$. Let D_w be an arbitrary database over the world-relations and D_s an arbitrary database over the site-relations, and let D'_s be $(D_s \cup v^c(D_w)) \cap v^l(D_w)$. Then D'_s satisfies $v^c(D_w) \subseteq D'_s \subseteq v^l(D_w)$. Because by theorem 1, r' is semantically correct it follows that $r'[s \mapsto s \wedge v^l](D_w, D_s) \subseteq r'(D'_s) \subseteq r(D'_s) \subseteq r[s \mapsto s \vee v^c](D_w, D_s)$. Therefore, $r'[s \mapsto s \wedge v^l]$ is contained in $r[s \mapsto s \vee v^c]$. For the opposite direction, assume $r'[s \mapsto s \wedge v^l]$ is contained in $r[s \mapsto s \vee v^c]$ for every retrievable query with $r'[s \mapsto v^l] \subseteq q$. Let r'' be a semantically correct retrievable query and let D_w and D_s be databases over the world- and site-relations respectively with $v^c(D_w) \subseteq D_s \subseteq v^l(D_w)$. By theorem 1, $r''[s \mapsto v^l] \subseteq q$ and therefore $r''(D_s) \equiv r''[s \mapsto s \wedge v^l](D_w, D_s) \subseteq r[s \mapsto s \vee v^c](D_w, D_s) \equiv r(D_s)$. Therefore, r is source-complete. □

Query plan optimization

In general, there will be infinitely many retrievable queries r' with $r'[s \mapsto v^l] \subseteq q$. It therefore seems as if the criterion for source-completeness is not effective. However, it is sufficient to only consider the finite number of conjunctive retrievable queries generated by the algorithm in, for example, (Qian 1996). Applied to a conjunctive query q and the liberal views $v_1^l, \ldots, v_n^l$, Qian's algorithm produces a set of conjunctive retrievable queries, denoted $folding(q, v^l)$, with the following properties:

(i) $r[s \mapsto v^l] \subseteq q$ for every $r \in folding(q, v^l)$.

(ii) For every conjunctive retrievable query r' with $r'[s \mapsto v^l] \subseteq q$, $r' \subseteq r$ for some $r \in folding(q, v^l)$.

By theorem 1, the first property guarantees that each conjunctive retrievable query in $folding(q, v^l)$ is semantically correct with respect to q. Therefore, the union of all conjunctive retrievable queries in $folding(q, v^l)$, denoted $\bigcup folding(q, v^l)$, is semantically correct. The second property states that for every

semantically correct conjunctive retrievable query r' and *every* database D_s over the site-relations, $r'(D_s)$ is contained in $\bigcup folding(q, v^l)(D_s)$. It follows that specifically for databases D_s satisfying $v^c(D_w) \subseteq D_s \subseteq v^l(D_w)$ for some database D_w over the world-relations, $r'(D_s)$ is contained in $\bigcup folding(q, v^l)(D_s)$. Therefore $\bigcup folding(q, v^l)$ is source-complete. The second property has a further implication. A query r' that is contained in a query r requires the same site-relations as r, and possibly more. It follows that there is a semantically correct, source-complete, and view-minimal retrievable query of the form $\bigcup_{i \in I} r_i$ with $r_i \in folding(q, v^l)$ for all $i \in I$.

Information brokers that do not have any local completeness information have to retrieve a query equivalent to $\bigcup folding(q, v^l)$ in order to guarantee source-completeness. However, $\bigcup folding(q, v^l)$ is in general not view-minimal. By using the local completeness information given by the conservative views, conjunctive queries can be removed from $folding(q, v^l)$ without losing source-completeness. The following theorem gives the crucial criterion for identifying the proper subset of $folding(q, v^l)$ that is both source-complete and view-minimal.

Theorem 3 *If $\bigcup_{j \in J} r_j$ is a semantically correct retrievable query that contains every semantically correct retrievable query, then for every $I \subseteq J$ satisfying*

$$\bigcup_{j \in J-I} r_j[s \mapsto s \wedge v^l] \subseteq \bigcup_{i \in I} r_i[s \mapsto s \vee v^c], \qquad (*)$$

the retrievable query $\bigcup_{i \in I} r_i$ is source-complete. Moreover, if I is chosen as the set minimizing the number of information sources required in $\bigcup_{i \in I} r_i$ and satisfying (*), then $\bigcup_{i \in I} r_i$ is view-minimal.

Proof. Because $\bigcup_{j \in J} r_j$ contains all semantically correct retrievable queries we have

$$\begin{aligned} r'(D_s) &\subseteq \bigcup_{j \in J} r_j(D_s) \\ &\equiv \bigcup_{j \in J-I} r_j[s \mapsto s \wedge v^l](D_w, D_s) \\ &\quad \cup \bigcup_{i \in I} r_i[s \mapsto s \vee v^c](D_w, D_s) \\ &\subseteq \bigcup_{i \in I} r_i[s \mapsto s \vee v^c](D_w, D_s) \\ &\equiv \bigcup_{i \in I} r_i(D_s) \end{aligned}$$

for all databases D_w and D_s over the world- and site-relations respectively with $v^c(D_w) \subseteq D_s \subseteq v^l(D_w)$. Therefore, $\bigcup_{i \in I} r_i$ is source-complete. Let I' be a subset of J such that $\bigcup_{i \in I'} r_i$ is semantically correct, source-complete, and view-minimal. Because I' satisfies condition (*) by theorem 2, $\bigcup_{i \in I} r_i$ requires the same number of information sources as $\bigcup_{i \in I'} r_i$. Therefore, $\bigcup_{i \in I} r_i$ is view-minimal. □

Theorem 3 immediately suggests an algorithm for finding semantically correct, source-complete and view-minimal retrievable queries given *conjunctive* user queries *without* built-in predicates. First, compute $folding(q, v^l)$, and then find a subset R of $folding(q, v^l)$ requiring the least number of site-relations such that $\bigcup(folding(q, v^l) - R)[s \mapsto s \wedge v^l]$ is contained in $\bigcup R[s \mapsto s \vee v^c]$. Then $\bigcup R$ is the desired retrievable query.

This special case can be easily generalized to handle unions of conjunctive queries with built-in predicates as user queries. User queries then are of the form

$$q \equiv \bigcup_{j \in J} (q_j^w \wedge q_j^b)$$

where the q_j^w's are conjunctive queries in terms of world-relations without built-in predicates, and the q_j^b's are conjunctive queries consisting only of built-in predicates. In this case,

$$\bigcup_{j \in J} (\bigcup folding(q_j^w, v^l) \wedge q_j^b)$$

is semantically correct with respect to q and contains all retrievable queries that are semantically correct with respect to q. Again, theorem 3 can be applied to find a subset R of the set of conjunctive retrievable queries in this union such that $\bigcup R$ is source-complete and view-minimal.

Related work

Levy (Levy 1996) uses local completeness information to test whether a query plan is complete, but doesn't consider query optimization. In (Kirk *et al.* 1995), Kirk et al. claim to have an algorithm that makes use of local completeness information and guarantees view-minimality. Their algorithm first determines the part of a user query that is known to be stored completely by some information sources. It selects a minimal set of information sources that provide this part of the query. In a second step, every information source that might contribute some data to the remaining part of the query is added. This algorithm doesn't guarantee view-minimality, however, as can be seen from the following counterexample. Consider three information sources IS_{new}, IS_{bmw}, and IS_{honda} that store fragments of the *bluebook* relation from example 2. IS_{new} stores all information for cars built in 1997, and nothing else. IS_{bmw} and IS_{honda} store information for cars built by BMW and Honda respectively. They are complete for information on cars built by BMW and Honda

respectively in 1997. Suppose a user requests current market values for cars built by BMW and Honda. The part of the query that is guaranteed to be stored is the fragment of the *bluebook* relation for the year 1997. IS_{new} alone guarantees to provide this fragment. For the remaining part of the query though, IS_{bmw} and IS_{honda} have to be included. Therefore, the query plan resulting from the algorithm in (Kirk *et al.* 1995) accesses all three information sources. However, this query plan is not view-minimal, because it is sufficient to only request information from the BMW and the Honda database.

Approximating a relation by two views is studied in the context of predicate caching (Keller & Basu 1996) and in relation to the question of whether datalog programs can be approximated by unions of conjunctive queries (Chaudhuri 1993; Chaudhuri & Kolaitis 1994). Our terminonolgy of "conservative" and "liberal" views is adopted from (Keller & Basu 1996). The work of Chaudhuri in (Chaudhuri 1993) and Chaudhuri and Kolaitis in (Chaudhuri & Kolaitis 1994) is of interest here because it points to limitations of the source-centric approach. If for example an information source stores the transitive closure of a predicate p, then there are no nonrecursive views that could be used as close approximations of this site-relation.

Conclusion

We considered the problem of query plan optimization in the source-centric approach to information integration. We showed how local completeness information can be used to avoid redundant accesses to information sources. Our algorithm proceeds in two steps. In the first step, a semantically correct and source-complete retrievable query plan is generated using one of the algorithms in (Qian 1996; Levy, Rajaraman, & Ordille 1996a; 1996b; Duschka & Genesereth 1997a; 1997b). In the second step, redundant parts of this retrievable query plan are eliminated. The resulting query plan is guaranteed to be semantically correct, source-complete, and view-minimal.

Acknowledgements

We would like to thank Whitney Carrico, Harish Devarajan, Michael Genesereth, and Alon Levy for helpful comments that improved this paper.

References

Chandra, A. K., and Merlin, P. M. 1977. Optimal implementation of conjunctive queries in relational data bases. In *Proc. 9th ACM STOC*, 77–90.

Chaudhuri, S., and Kolaitis, P. G. 1994. Can datalog be approximated? In *Proc. 13th ACM PODS.*

Chaudhuri, S. 1993. Finding nonrecursive envelopes for datalog predicates. In *Proc. 12th ACM PODS.*

Chawathe, S.; Garcia-Molina, H.; Hammer, J.; Ireland, K.; Papakonstantinou, Y.; Ullman, J.; and Widom, J. 1994. The TSIMMIS project: Integration of heterogeneous information sources. In *Information Processing Society of Japan*, 7–18.

Duschka, O. M., and Genesereth, M. R. 1997a. Answering recursive queries using views. In *Proc. 16th ACM PODS.*

Duschka, O. M., and Genesereth, M. R. 1997b. Query planning in Infomaster. In *Proc. ACM SAC.*

Duschka, O. M., and Levy, A. Y. 1997. Recursive plans for information gathering. In *Proc. 15th IJCAI.*

Etzioni, O.; Golden, K.; and Weld, D. 1994. Tractable closed world reasoning with updates. In *Proc. 4th KR*, 178–189.

Geddis, D. F.; Genesereth, M. R.; Keller, A. M.; and Singh, N. P. 1995. Infomaster: A virtual information system. In *Intelligent Information Agents Workshop at CIKM '95.*

Keller, A. M., and Basu, J. 1996. A predicate-based caching scheme for client-server database architectures. *The VLDB Journal* 5:35–47.

Kirk, T.; Levy, A. T.; Sagiv, Y.; and Srivastava, D. 1995. The Information Manifold. In *Proc. AAAI Symp. on Information Gathering in Distributed Heterogeneous Environments.*

Klug, A. 1988. On conjunctive queries containing inequalities. *J. ACM* 35(1):146–160.

Levy, A. Y. 1996. Obtaining complete answers from incomplete databases. In *Proc. 22nd VLDB*, 402–412.

Levy, A. Y.; Mendelzon, A. O.; Srivastava, D.; and Sagiv, Y. 1995. Answering queries using views. In *Proc. 14th ACM PODS.*

Levy, A. Y.; Rajaraman, A.; and Ordille, J. J. 1996a. Query-answering algorithms for information agents. In *Proc. 13th AAAI Conference*, 40–47.

Levy, A. Y.; Rajaraman, A.; and Ordille, J. J. 1996b. Querying heterogeneous information sources using source descriptions. In *Proc. 22nd VLDB*, 251–262.

Qian, X. 1996. Query folding. In *Proc. 12th ICDE.*

Rajaraman, A.; Sagiv, Y.; and Ullman, J. D. 1995. Answering queries using templates with binding patterns. In *Proc. 14th ACM PODS.*

Sagiv, Y., and Yannakakis, M. 1980. Equivalence among relational expressions with the union and difference operators. *J. ACM* 27(4):633–655.