Estimating Soundness and Completeness of Queries Over Description Logic Ontologies *

Martin Peim, Enrico Franconi, Norman W. Paton
Dept. of Computer Science, Univ. of Manchester
Oxford Rd, Manchester M13 9PL, UK
lastname@cs.man.ac.uk

Abstract

Ontology-based information integration systems allow users to express queries over high-level conceptual models. However, such queries must subsequently be evaluated over collections of sources, some of which are likely to be expensive to use or subject to periods of unavailability. As such, it would be useful if information integration systems were able to provide users with estimates of the consequences of omitting certain sources from query execution plans. This paper presents an approach to estimating the soundness and completeness of queries expressed in the $\mathcal{ALCQI}$ description logic.

1 Introduction

In recent years, a number of distributed query-processing systems have been developed in which the global schema and user queries are expressed in some form of Description Logic (DL) (for example TAMBIS [6], DWQ [2], Information Manifold [9], PICSEL [5], SIMS [1]). The use of DLs as high-level data description languages has several advantages, including the use of reasoning support for DLs to assist in query formulation at the user-interface level and in query translation and optimisation.

Typically in these these systems, the first stage of query processing is to rewrite user queries expressed over some DL ontology into DL expressions using only atoms that can be mapped to source databases where each source contains a set of instances of some atomic DL concept or role. This mapping of sources to concepts and query rewriting may be done using either a *global-as-view* (e.g. SIMS, TAMBIS) or a *local-as-view* approach (e.g. DWQ, Information Manifold, PICSEL). Since our methods are applied to the rewritten query, they could be applied (with suitable modifications to adapt them to different DLs) to either type of system.

For some concepts and roles there may be several sources available to provide extents. The user of the system may wish to use only some of the available sources to reduce costs or download times. Furthermore, some sources may be unavailable at

*This work has been supported by the UK Engineering and Physical Science Research Council (EPSRC), whose support we are pleased to acknowledge

any given time, and the user may need to decide whether to wait for all sources to be available or to proceed with the query using the sources that are currently available. In order to evaluate such trade-offs, we need to be able to estimate the amount of information lost by only using a subset of the existing data, and the effects of such information loss on the quality of query answers. It is this problem that is addressed in this paper

If the DL query language contains non-monotonic operations such as negation and universal quantification (under a closed-world assumption), the omission of source data can lead to the inclusion of incorrect answers in the query answer set, as well as the exclusion of correct answers. One therefore needs to be concerned about both *incompleteness* ("how many of the answers to the query do we fail to retrieve?") and *unsoundness* ("how many of the answers we retrieve are incorrect?") of query plans.

The notion of completeness as a quality measure in distributed query planning is considered in [11] (our notion of *completeness* corresponds to their notion of *relevancy*). However, the query language considered there is much simpler than a DL. For example, it has no negation, so that soundness is not an issue, and the only merge operation in query planning is *join*. In [12], we describe a (global-as-view) distributed DL query processing system (with multiple sources for each source atom) in which queries are formulated in a relatively expressive DL, $\mathcal{ALCQI}$. This paper presents a method for estimating soundness and completeness of $\mathcal{ALCQI}$ query plans, by estimating the cardinality of the extents of associated $\mathcal{ALCQI}$ expressions (see [4] for an account of the estimation the cardinalities of intermediate results in relational algebra expressions in the context of database query optimisation). Our methods make use of statistical information which can be gathered from the source databases in advance of query processing (for comparison, in [10], the authors use information from database profiles and statistical sampling to estimate cardinalities of derived relations in database query processing).

Section 2 contains a brief account of query processing over DLs, in particular $\mathcal{ALCQI}$. Section 3 presents our definitions of soundness and completeness for query plans. Section 4 describes our estimation method for concept cardinalities. Section 5 presents some results of statistical testing of our methods. Finally, Section 6 contains some conclusions and suggestions for further work.

2 Query answering over the Description Logic $\mathcal{ALCQI}$

The basic types of a DL are *concepts* and *roles*. A concept expression is a description gathering the common properties among a collection of individuals; from a logical point of view it is a unary predicate ranging over the domain of individuals. Interrelationships between these individuals are represented by means of role expressions (which are interpreted as binary relations over the domain of individuals). Roles can be seen as denoting properties of individuals, since they associate values for the property to individuals of a given class.

$\mathcal{ALCQI}$ is a description logic featuring a rich combination of constructors, including full boolean operators, qualified number restrictions, inverse roles and general inclusion assertions. The syntax rules below define valid concept and role expressions.

$$C, D \to A \mid \top \mid \bot \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \forall R.C \mid \exists R.C \mid \exists^{\geq n} R.C \mid \exists^{\leq n} R.C$$
$$R \to P \mid R^{-}$$

Here C, D and R denote concept and role expressions, A is an atomic concept name and P is an atomic role name. The semantics are defined in the usual compositional fashion in terms of an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a (non-empty) domain, containing every object in the world under consideration and $\cdot^{\mathcal{I}}$ is an interpretation function. For details see [3], for example.

A DL *ontology* consists of a set of concept and role names and a set of axioms, or constraints, expressed in terms of those names. The axioms express equality or containment (also known as *subsumption*) between role and concept expressions. They limit the set of admissible interpretations for the ontology to those which satisfy the axioms. Since our estimation methods do not take axioms into account, we omit any further details. A more sophisticated treatment of cardinality estimation might be able to make use of information contained in the axioms.

A *query* over an $\mathcal{ALCQI}$ ontology is simply a concept expression. The *answer* to a query (with respect to a model for the ontology) is simply the set of instances of that expression. We restrict our attention to *safe* queries, where a query is considered safe if answering that query does not involve looking up information not referred to in the query. This is crucial to restrict the scope of a query. For example, the query $\forall R.C$ is unsafe because answering it involves, among other things, finding all individuals with no R-fillers, and this information cannot be obtained by examining the instances of R and C. To check if a query Q is safe we rewrite it into a negation normal form[1] Q'. Then Q is safe if it has the form $\bot$, A (where A is atomic), $\exists R.C$ or $\exists^{\geq n} R.C$ $(n \geq 1)$. It is unsafe if it has the form $\top$, $\neg A$, $\forall R.C$ or $\exists^{\leq n} R.C$. A conjunction is safe if and only if at least one of its conjuncts is safe. A disjunction is safe if and only if all of its disjuncts are safe. Note that, under this definition, a concept expression is safe if and only if its negation is unsafe.

We present below a proposal for estimating soundness and completeness of query plans. The measures we use are based on estimations of the cardinalities of concept extensions. We assume throughout that we have a fixed $\mathcal{ALCQI}$ ontology B and a fixed (finite) interpretation $\mathcal{I}$ of B. For any concept expression C over the symbols of B we define the *cardinality* of C, denoted $\#(C)$, to be the number of elements of its extension $C^{\mathcal{I}}$.

3 Soundness and Completeness

In order to assess trade-offs between soundness and completeness of query plans and factors such as cost and availability of data sources, we need quantitative measures of soundness and completeness. In this section we define a pair of such measures. They

[1] By pushing negations inwards in the usual way, one can rewrite any $\mathcal{ALCQI}$concept expression into an equivalent expression in *negation normal form* or NNF, where negations only appear in front of concept names.

are based on the cardinalities of DL concepts. We will express our definitions in terms of the $\mathcal{ALCQI}$ DL, but similar definitions could be made for other DLs.

We assume that all queries are expressed in terms of a set of basic concept and role names (atoms) and that for each basic atom we have a set of sources available. In a complete query-processing system, we would have other concept names in the ontology and a mechanism for rewriting user queries into queries involving only the basic names. Each source for a concept C is represented by a concept name C' which is *subsumed* by C. That is, in any admissible model for the ontology, each instance of the C' is an instance of C. Similarly, each role source for a role R is a role R' which is subsumed by R. The representing concepts and roles will be referred to as *source concepts* and *source roles*.

For the purposes of this document, a query plan is simply a selection from the available set of sources for each role and concept name in the query; only the selected sources are to be used to answer the query. Other, lower-level, choices (such as the order in which sources are accessed) may be made in the query-planning process, but we assume that such choices do not affect the set of concept instances which are returned by the plan, and so do not affect soundness or completeness. More formally, a *plan* P for an $\mathcal{ALCQI}$ query Q is an $\mathcal{ALCQI}$ expression in which the atomic concepts and roles in Q have been replaced by unions of smaller subconcepts and subroles:

Definition 1 *Let Q be a query (that is, an $\mathcal{ALCQI}$ expression). A* plan *P for Q is a query that can be obtained from Q by substituting a concept expression C' for each occurrence of an atomic concept C in Q, and a role R' for each occurrence of a role R in Q, subject to the following restrictions:*

- *For each C, C' is a union $C'_1 \sqcup \ldots \sqcup C'_k$, where each C_k is an atomic concept whose instances are contained in those of C.*
- *For each R, R' is a role whose instances are contained in those of R.*

Our notions of soundness and completeness are analogous to the notions of precision and recall, respectively in the field of Information Retrieval. The *soundness* of a plan P with respect to a given query Q, denoted $\mathcal{S}(P,Q)$, is a measure of how many of the answers it produces are in fact members of the set of instances defined by the query. Although the definition given below depends on the contents of the databases, we will suppress the database state from the notation, and take it as reads that we have in mind a particular database state.

Definition 2 *Let P be a plan for a query Q. The* soundness *of P for Q, denoted S(P, Q) is:*

$$\mathcal{S}(P,Q) = \#(P \sqcap Q)/\#(P). \tag{1}$$

Note that soundness always takes values between 0 and 1. Furthermore, the soundness of an empty plan (one which produces no answers) is not defined by the above formula. Since an empty plan clearly produces no wrong answers, we adopt the convention that the soundness of such a plan is 1.

Similarly, the *completeness* $\mathcal{C}(P,Q)$ of a plan P with respect to a query Q is a measure of how many of the answers to Q are actually found by P.

Definition 3 *Let P be a plan for a query Q. The* completeness *of P for Q, denoted C(P, Q) is:*

$$\mathcal{C}(P, Q) = \#(P \sqcap Q)/\#(Q). \tag{2}$$

Again, we note that completeness takes values in the range $[0, 1]$. In this case, the formula cannot be applied when the *query* is empty. Since we cannot miss any answers to an empty query, we assign a value of 1 in this case also.

So we can measure (or estimate) the soundness and completeness of a plan if we can measure (or estimate) the cardinality of an arbitrary concept expression.

4 Estimating Cardinalities of DL Concept Extensions

4.1 Propositional Expressions

We begin by considering the propositional fragment of $\mathcal{ALCQI}$, where the only connectives used are $\sqcap$, $\sqcup$ and $\neg$. This fragment is equivalent to classical propositional logic. In order to compute the cardinality of any safe propositional expression, it suffices to know the cardinalities of all conjunctions of atomic concepts. We can obtain them from the databases in advance of any query processing. Since the number of such conjunctions is exponential in the number of atomic concepts, it may not be feasible to collect and store the cardinalities of all of them, except in cases where most of the cardinalities are zero. For example, it might be that the concept atoms can be divided into small "clusters" such that only concepts in the same cluster can have non-empty intersections. In such cases, we must choose a data structure and lookup algorithm so that the empty cases can be detected efficiently. In the general case, we would expect the cardinalities of conjunctions of many concepts to be small compared to those of few concepts. So we could compute approximate cardinalities for propositional expressions (a least, for those involving only short conjunctions) by taking the cardinalities of long conjunctions to be zero. Here the length of a "long" conjunction would have to be determined by an analysis of the data.

Given such input, we can compute the cardinalities of all safe propositional concept expressions by (recursively) using the equations

$$\#(C \sqcup D) = \#(C) + \#(D) - \#(C \sqcap D) \tag{3}$$

$$\#(C \sqcap \neg D) = \#(C) - \#(C \sqcap D) \tag{4}$$

It may be noted that for large concept expressions the application of equations (3) and (4) could involve a large amount of computation. For example, calculating the cardinality of a union $\bigcup_{i=1}^{n} C_i$ of atomic concepts C_i by rule (3) involves a summation of $2^n - 1$ terms (one for each non-empty subset of $\{1, \ldots, n\}$):

$$\#(\bigcup_{i=1}^{n} C_i) = \sum_{\{i_1,\ldots,i_k\} \subseteq \{1,\ldots,n\}} (-1)^k \, \#(C_{i_1} \cap \ldots \cap C_{i_k}). \tag{5}$$

This formula, which is easily proved by induction on n, is the well-known inclusion-exclusion principle of combinatorics (see [8] pages 178–179, for example). However, in practice it is likely to be the case that most of the intersections are empty. In such cases much of the recursion may be avoided by not evaluating the (necessarily zero) last term on the right hand side of equation (3) if either of the first two terms is zero. Similarly, we should not evaluate the second term on the right hand side of equation (4) if the first term is zero.

As an example, suppose we have concepts C_1, represented by a single source S_1, and C_2, represented by S_{21} and S_{22}. Suppose also that the sources taken together provide all instances of C_1 and C_2 so that we have equivalences $C_1 \equiv S_1$ and $C_2 \equiv S_1 \sqcup S_2$. Let the cardinalities of the sources and their intersections be given as in table 1. Consider

Concept	Cardinality
S_1	1000
S_{21}	1000
S_{22}	1000

Concept	Cardinality
$S_1 \sqcap S_{21}$	500
$S_1 \sqcap S_{22}$	100
$S_{21} \sqcap S_{22}$	0
$S_1 \sqcap S_{21} \sqcap S_{22}$	0

Table 1: Cardinalities for example sources

the query $Q_2 = C_1 \sqcap \neg C_2$ and the plan $P_2 = S_1 \sqcap \neg S_2$. Since the incomplete data is is used for a concept which is under a negation, we have $P_2 \sqcap Q_2 \equiv Q_2$ so that $\mathcal{C}(P_2, Q_2) = 1$. For soundness, we have

$$\begin{aligned}
\#(Q_2) &= \#(S_1 \sqcap \neg(S_{21} \sqcup S_{22})) \\
&= \#(S_1) - \#(S_1 \sqcap (S_{21} \sqcup S_{22})) \\
&= \#(S_1) - \#((S_1 \sqcap S_{21}) \sqcup (S_1 \sqcap S_{22})) \\
&= \#(S1) - \#(S_1 \sqcap S_{21}) - \#(S_1 \sqcap S_{22}) + \#(S_1 \sqcap S_{21} \sqcap S_{22}) \\
&= 1000 - 500 - 100 + 0 = 400
\end{aligned}$$

and $\#(P_2) = \#(S_1 \sqcap \neg S_{21}) = \#(S_1) - \#(S_1 \sqcap S_{21}) = 1000 - 500 = 500$, which gives $\mathcal{S}(P_2, Q_2) = \#(P_2 \sqcap Q_2)/\#(P_2) = \#(Q_2)/\#(P_2) = 400/500 = 0.8$.

4.2 Quantified expressions

In order to estimate the cardinalities of quantified formulae, we adopt a probabilistic approach. As in the case of propositional expressions, the data required for our calculations can (in principle) be obtained by running a finite (though possibly large) number of queries in advance of general query processing. However, we cannot expect to obtain precise cardinality results, even with precise input data, since the filler concept C in an expression such as $\exists R.C$ may be an arbitrary complex concept expression. The data we require concerns the distributions of the numbers of fillers for elements in the domain of each role. First, we assume that we have concept expressions domain(R) and range(R) for the domain and range of any role R. These should be unions of atomic concepts (in an application, the atomic concepts will be the concepts domains and ranges of the corresponding database relations). The best results are likely to be obtained if the expression range(R) accurately captures the actual range of filler values of R, rather than just the potential values. That is, if there are not too many instances y of R with no x such that (x, y) is an instance of R. For each $i \geq 0$ Let q_i be the probability that an element of domain(R) has exactly i R-fillers(so $\sum_{i=0}^{\infty} q_i = 1$). That is,

$$q_i = (\#(\exists^{\geq n} R.\top) - \#(\exists^{\geq n-1} R.\top))/\#(\mathrm{domain}(R)). \tag{6}$$

As with the cardinalities of intersections of source concepts, we could evaluate the q_i for each role in advance of query processing, and update the values periodically. If the q_i are calculated directly form equation (6), then only finitely many of them are non-zero for each R. In fact, if R is single-valued then $q_i = 0$ for $i > 1$. I general, we require that this finiteness condition holds for any set of values which we use for the q_i. (In some situations, we might approximate the q_i by, say, a Poisson distribution with a given mean, suitably truncated.)

To estimate the cardinality of an existentially quantified formula, $\exists^{\geq n} R.C$, where $n \geq 1$, let $P(n, R, C)$ be the probability that an element of domain(R) has at least n R-fillers in C. Then

$$\#(\exists^{\geq n} R.C) = \#(\mathrm{domain}(R))P(n, R, C), \tag{7}$$

so we need to estimate $P(n, R, C)$.

Let p be the probability that any given R-filler is in C. We can estimate p by the formula

$$p \approx \#(\mathrm{range}(R) \sqcap C)/\#(\mathrm{range}(R)). \tag{8}$$

(we can assume that $\#(\mathrm{range}(R)) \neq 0$ since otherwise we have $P(n, R, C) = 0$ for any C). Note that the expression range(R) $\sqcap C$ is safe (even if C is unsafe) and contains fewer quantifiers than $\exists^{\geq n} R.C$ so, by induction, we can estimate its cardinality. Then

$$P(n, R, C) = \sum_{i=1}^{\infty} q_i(\text{probability that at least } n \text{ out of } i \text{ fillers are in } C). \tag{9}$$

Since only finitely many of the q_i are non-zero for each R, the sum in equation (9) is actually finite.

Now let us assume that the probabilities of distinct R-fillers (for a given object) being in C are independent of each other. Then, given that we have exactly i fillers, the probability that j of them are in C is given by a binomial distribution. The probability that j out of i fillers are in C is $\binom{i}{j} p^j (1-p)^{i-j}$, where $\binom{i}{j}$ denotes the binomial coefficient $i!/(j!(i-j)!)$. This formula derives from the fact that there are $\binom{i}{j}$ j-element subsets of the i fillers and, for each subset, the probability that all of its elements and none of the other fillers are in C is $p^j(1-p)^{i-j}$. Then we can estimate $P(n, R, C)$ by $Q(n, R, C)$ where

$$Q(n, R, C) = \sum_{i=1}^{\infty} q_i \sum_{j=n}^{i} \binom{i}{j} p^j (1-p)^{i-j}. \tag{10}$$

Note that when $i \geq 2n$, the calculation of the inner sum can be simplified by using the identity

$$\sum_{j=n}^{i} \binom{i}{j} p^j (1-p)^{i-j} = 1 - \sum_{j=0}^{n-1} \binom{i}{j} p^j (1-p)^{i-j}, \tag{11}$$

where the sum on the right hand side has fewer terms than the one on the left. (Equation (11) is easily derived from the fact that the two summations taken together form the binomial expansion of $(p + (1-p))^i$). In particular, if $n = 1$ (so we are considering the concept $\exists R.C$), we can compute $Q(1, R, C)$ as

$$Q(1, R, C) = \sum_{i=1}^{\infty} q_i (1 - (1-p)^i). \tag{12}$$

In general, if q_i is non-zero for large i we can approximate the inner sum in equation (10) for large values of i by using a normal (Gaussian) distribution with mean $\mu = ip$ and standard deviation $\sigma = \sqrt{ip(1-p)}$:

$$\sum_{j=n}^{i} \binom{i}{j} p^j (1-p)^{i-j} \approx \frac{1}{\sqrt{2\pi}} \int_{(n-\mu)/\sigma}^{\infty} e^{-\frac{1}{2}x^2}\, dx. \tag{13}$$

The value of this integral can be approximated by interpolation in standard statistical tables.

So we can estimate $\#(\exists^{\geq n} R.C)$ by the formula

$$\#(\exists^{\geq n} R.C) \approx \#(\text{domain}(R)) Q(n, R, C). \tag{14}$$

More generally, let D be a concept expression, and suppose we wish to estimate $\#(D \sqcap \exists^{\geq n} R.C)$. If we assume that restricting attention to instances of D does not affect either the distribution of numbers of R-fillers (the q_i defined above) or the probability p that a given R-filler is in C, we have the approximation

$$\#(D \sqcap \exists^{\geq n} R.C) \approx \#(D \sqcap \text{domain}(R)) Q(n, R, C). \tag{15}$$

Note that the expression $D \sqcap \text{domain}(R)$ is safe, so that we can estimate its cardinality, provided the expression for domain(R) is safe.

Similarly, suppose we wish to estimate $\#(D \sqcap \exists^{\geq n_1} R_1.C_1 \sqcap \exists^{\geq n_2} R_2.C_2)$. We assume that as above that membership in D does not influence the values of p or the q_i for either R_1 or R_2, and that R_1 and R_2 do not influence each other. Then we use the approximation

$$\begin{aligned} &\#(D \sqcap \exists^{\geq n} R_1.C_1 \sqcap \exists^{\geq n} R_2.C_2) \approx \\ &\qquad \#(D \sqcap \text{domain}(R_1) \sqcap \text{domain}(R_2)) Q(n_1, R_1, C_1) Q(n_2, R_2, C_2) \end{aligned} \tag{16}$$

and so on.

By combining rules like (15) and (16) with rules (3) and (4) we can estimate the cardinality of any safe $\mathcal{ALCQI}$ concept expression.

Domain and range	
Concept	Cardinality
$S_1 \sqcap \text{domain}(R)$	800
$\text{range}(R)$	1500
$S_{21} \sqcap \text{range}(R)$	600
$S_{22} \sqcap \text{range}(R)$	400
$S_{21} \sqcap S_{22} \sqcap \text{range}(R)$	0

Filler count distribution	
i	q_i
0	0.35
1	0.37
2	0.18
3	0.08
4	0.02
≥ 5	0

Table 2: Cardinality data for the example role R

Consider the example of Section 4.1. Suppose we have a role R with a single source S_R, which we assume to provide complete information for R, so $S_R \equiv R$. The data we need concerning the role R are contained in table 2. The query $Q_3 = C_1 \sqcap \exists^{\leq 1} S_R.C_2$ admits a plan $P_3 = S_1 \sqcap \exists^{\leq 1} R.S_{21}$. Then, as with the example involving Q_2, we have $P_3 \sqcap Q_3 \equiv Q_3$ and so $\mathcal{C}(P_3, Q_3) = 1$. Since, in general, $\exists^{\leq n} R.C \equiv \neg\exists^{\geq n+1} R.C$, we have

$$\begin{aligned}
\#(Q_3) &= \#(S_1 \sqcap \neg\exists^{\geq 2} R.(S_{21} \sqcup S_{22})) \\
&= \#(S_1) - \#(S_1 \sqcap \exists^{\geq 2} R.(S_{21} \sqcup S_{22})) \\
&\approx \#(S_1) - \#(S_1 \sqcap \text{domain}(R)) Q(2, R, S_{21} \sqcup S_{22})
\end{aligned}$$

To evaluate $Q(2, R, S_{21} \sqcup S_{22})$, we need to compute

$$\begin{aligned}
p &= \frac{\#(\text{range}(R) \sqcap S_{21} \sqcup S_{22})}{\#(\text{range}(R))} \\
&= \frac{\#(\text{range}(R) \sqcap S_{21}) + \#(\text{range}(R) \sqcap S_{22}) - \#(\text{range}(R) \sqcap S_{21} \sqcap S_{22})}{\#(\text{range}(R))} \\
&= (600 + 400 - 0)/1500 = 2/3.
\end{aligned}$$

Then

$$\begin{aligned}
Q(2, R, S_{21} \sqcup S_{22}) &= q_2 \binom{2}{2} p^2 + q_3 \left(\binom{3}{2} p^2 (1-p) + \binom{3}{3} p^3 \right) \\
&\quad + q_4 \left(1 - \binom{4}{0} (1-p)^4 - \binom{4}{1} p (1-p)^3 \right) \\
&= 0.18(1(4/9)) + 0.08(3(4/27) + 1(8/27)) + .02(1 - 1(1/81) - 4(2/81)) \\
&\approx 0.16,
\end{aligned}$$

so our approximation to $\#(Q_3)$ is

$$\#(Q_3) \approx 1000 - 800 \times 0.16 \approx 870.$$

For $\#(P_3)$ we make a similar calculation for $Q(2, R, S_{21})$, except that we use the value

$$\frac{\#(\text{range}(R) \sqcap S_{21})}{\#(\text{range}(R))} = 600/1500 = 2/5$$

for p. This yields the value 0.056. So

$$\begin{aligned}\#(P_3) &= \#(S_1 \sqcap \neg\exists^{\geq 2} R.S_{21}) = \#(S_1) - \#(S_1 \sqcap \exists^{\geq 2} R.S_{21}) \\ &\approx \#(S_1) - \#(S_1 \sqcap \mathrm{domain}(R))Q(2, R, S_{21}) \\ &\approx 1000 - 800 \times 0.056 \approx 960.\end{aligned}$$

So our estimate of the soundness is $\mathcal{S}(P_3, Q_3) = \frac{\#(P_3 \sqcap Q_3)}{\#(P_3)} = \frac{\#(Q_3)}{\#(P_3)} \approx \frac{870}{960} \approx 0.91$.

5 Statistical Evaluation of Estimation Methods

In order to validate the method proposed in Section 4.2 for estimating cardinalities of quantified expressions and, by extension, for estimating soundness and completeness of evaluation plans for such expressions, we have run some statistical tests using randomly-generated data.

We consider the expression $\exists R.C$, which is equivalent to $\exists^{\geq 1} R.C$. For given values of $d = \#(\mathrm{domain}(R))$, $r = \#(\mathrm{range}(R))$, and $s = \#(R)$ the test system generates a random s-element subset of $\{1 \ldots d\} \times \{1 \ldots r\}$, to represent the instances of R. This is done by using uniformly distributed random numbers for the domain and range values until s distinct pairs have been generated. The system also generates two "sources" C_1 and C_2 for the filler relation C by generating two random subsets of {1 ...r} with specified cardinalities c_1 and c_2. We then use our estimation techniques to compute estimates of $\#(C_1)$ and of the completeness $\mathcal{C}(C_1, C)$ of C_1 with respect to $C = C_1 \sqcup C_2$. We have examined the behaviour of our estimation techniques for varying values of the specified cardinalities.

Table 3 indicates the accuracy of our cardinality estimations. It contains data for

$\#(C_1)$	$\#(C_2)$	Mean of $\#(\exists R.C_1)$	Normalised error
10	50	9.42	0.29
20	50	18.26	0.18
30	50	25.78	0.14
40	50	33.18	0.11
50	50	39.47	0.09
60	50	45.20	0.07
70	50	50.62	0.06
80	50	55.36	0.05
90	50	59.59	0.03

Table 3: Cardinality estimation for 1000 trials with $\#(\mathrm{domain}(R)) = \#(\mathrm{range}(R)) = \#(R) = 100$

varying values of $\#(C_1)$ for the case where $\#(\mathrm{domain}(R))$, $\#(\mathrm{range}(R))$ and $\#(R)$ have all been fixed at 100. Results are shown for 1000 trials for each value of $\#(C_1)$. The table shows the mean of the actual values of $\#(\exists R.C_1)$. As a measure of the accuracy of our methods in estimating $\#(\exists R.C_1)$, the table shows a *normalised error* value. The normalised error is calculated by taking the root mean square (RMS) of the differences between the estimated and true values and dividing by by the mean of all the true values, so that it represents a relative, rather than absolute, error.

Table 4 shows results for completeness estimations, using the same generated data

$\#(C_1)$	$\#(C_2)$	Mean completeness	RMS error
10	50	0.26	0.07
20	50	0.40	0.07
30	50	0.53	0.07
40	50	0.65	0.07
50	50	0.75	0.06
60	50	0.82	0.05
70	50	0.88	0.04
80	50	0.92	0.03
90	50	0.97	0.02

Table 4: Completeness estimation with $\#(\text{domain}(R)) = \#(\text{range}(R)) = \#(R) = 100$

as in Table 3 with the addition of a set of instances for C_2, which in these examples always has cardinality 50. The table shows the mean of the true values of the completeness $\mathcal{C}(\exists R.C_1, \exists R.C)$ and the RMS deviation of the estimated value from the true value.

We observe from these tables (and from further test results which have been omitted to save space) that our estimation methods appear to be reasonably successful in cases where the cardinalities are not too small, but tend to break down for small answer sets. This is what one might expect. For example, if the cardinality of a set is estimated as 0.2, then in most cases the true figure will be either 0 or 1, and which one it is will make a difference which is more significant than the difference between, say 99 and 100. In the context of our query processing system, we may be able to improve the accuracy of our soundness and completeness estimates by evaluating subqueries which we estimate to have a small number of elements. If the access times for the sources involved in the subqueries have a latency which is not too high, we will be able to run such subqueries quickly in most cases. The size threshold below which we would use this technique would be determined by experiment.

6 Conclusions

DL based global conceptual models have been widely proposed for use in information integration systems. However, it is not always practical or possible for queries over such models to access all the extensional data in remote sources. Thus it is important that query processing environments are able to provide indications to users of the consequences for the soundness and completeness of results when specific sources are omitted. The methods presented in this paper represent a direct approach to providing such indications.

The results of the statistical validation tests (Section 5) suggest that our methods give good results in cases where the cardinality of the result is not too small, at least in the case of random data. However, we need to make further tests against real data and ontologies. We will address this issue as soon as the system described in [12] has been fully implemented.

Further work which could be done to improve our estimation technique might be to investigate the incorporation of statistical/probabilistic information about relations between cardinalities of concepts into the ontology, along the lines described in [7], to see whether this information could be used to improve our cardinality estimates.

References

[1] Y. Arens, C. A. Knoblock, and W-M. Shen. Query reformulation for dynamic information integration. *Journal of Intelligent Information Systems*, 6(2/3):99–130, 1996.

[2] D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, and R. Rosati. Information integration: Conceptual modeling and reasoning support. In *Proceedings of the 3rd IFCIS International Conference on Cooperative Information Systems*, pages 280–291. IEEE-CS Press, 1998.

[3] D. Calvanese, M. Lenzerini, and D. Nardi. Description logics for conceptual data modeling. In J. Chomicki and G. Saake, editors, *Logics for Databases and Information Systems*, pages 229–263. Kluwer Academic Publishers, 1998.

[4] H. Garcia-Molina, J. D. Ullman, and J. Widom. *Database System Implementation*. Prentice Hall, 2000.

[5] F. Goasdoué, V. Lattes, and M-C. Rousset. The use of Carin language and algorithms for information integration: the Picsel system. *International Journal of Cooperative Information Systems*, 9(4):383–401, 2000.

[6] C. A. Goble, R. Stevens, G. Ng, S. Bechhofer, N. W. Paton, P. G. Baker, M. Peim, and A. Brass. Transparent access to multiple bioinformatics information sources. *IBM Systems Journal*, 40(2):532–551, 2001.

[7] J. Heinsohn. A hybrid approach for modeling uncertainty in terminological logics. Technical report, Deutsches Forschungszentrum für Künstliche Intelligenz, 1991.

[8] D. E. Knuth. *The Art of Computer Programming, Vol. 1: Fundamental Algorithms*. Addison-Wesley, 2nd edition, 1973.

[9] A. Y. Levy, D. Srivastava, and T. Kirk. Data model and query evaluation in global information systems. *Journal of Intelligent Information Systems*, 5(2):121–143, 1995.

[10] F. Najjar and Y. Slimani. Cardinality estimation of distributed join queries. In *Proc. 10th DEXA Workshop*, pages 66–70. IEEE, 1999.

[11] F. Naumann, U. Leser, and J. C. Freytag. Quality-driven integration of heterogeneous information sources. In *Proc. 25th VLDB*, pages 447–458, Edinburgh, Scotland, 1999.

[12] M. Peim, E. Franconi, N. W. Paton, and C. A. Goble. Query processing with description logic ontologies over object-wrapped databases. Unpublished technical report, University of Manchester (`http://www.cs.man.ac.uk/~mpeim/query_processing.pdf`), 2001.