MIT/LCS/TM-229

THE COMPLEXITY OF EVALUATION RELATIONAL QUERIES

Stavros S. Cosmadakis

August 1982

THE COMPLEXITY OF EVALUATING RELATIONAL QUERIES

Stavros S. Cosmadakis

Laboratory for Computer Science

M.I.T., Cambridge, U.S.A

Abstract

We show that, given a relation R, a relational query φ involving only projection and join, and a conjectured result r, testing whether $\varphi(R)=r$ is D^p-complete. Bounding the size of $\varphi(R)$ from below (above) is NP-hard (co-NP-hard), and bounding it both ways is D^p-hard. Computing the size of $\varphi(R)$ is $\#P$-hard.

We also show that, given two relations R_1 and R_2 and two queries φ_1 and φ_2 as above, testing whether $\varphi_1(R_1)\subseteq\varphi_2(R_2)$ and testing whether $\varphi_1(R_1)=\varphi_2(R_2)$ are both Π_2^p-complete, even when $R_1=R_2$ or when $\varphi_1=\varphi_2$.

Keywords: Relational database, relational expression, polynomial transformation, completeness, hardness, NP, co-NP, D^p, enumeration problem, polynomial-time hierarchy.

This work was supported (in part) by the National Science Foundation under Grant No. MCS 81-20181.

1. Introduction

The relational model for databases [3] has been proposed as a formal means to describe data organization and to express queries. In this paper we are using the relational model to formalize the problem of evaluating a database query and the related problem of computing (or estimating) the size of the result, and we are using complexity theory to characterize the complexity of these problems. Specifically, we prove the following:

(i) Given a database s, a relational query Q and a conjectured result r, testing whether $Q(s)=r$ is complete for the class D^P (the class of lanquages which are equal to the intersection of a language in NP and a lanquage in co-NP --see [10, 6] for the relevant definitions). The problem is known to be NP-complete if we replace "=" by "$\supseteq$" [16], and co-NP-complete if we replace "=" by "$\subseteq$" [9]; we also give direct proofs of these facts.

(ii) Given a database s, a query Q and two non-negative integers d_1, d_2, testing whether $d_1 \leq |Q(s)| \leq d_2$ (where $|Q(s)|$ is the number of tuples of $Q(s)$) is D^P-hard, even when $d_1 = d_2$ or when $d_1 < d_2$. Testing whether $d_1 \leq |Q(s)|$ is NP-hard, and testing whether $|Q(s)| \leq d_2$ is co-NP-hard (the co-NP-hardnesss result also follows from the co-NP-completeness result of [9]).

(iii) Given a database s and a query Q, the enumeration problem of counting the number of tuples in the result $Q(s)$ is $\#P$-hard (see [13, 6] for relevant definitions).

We also prove the following results, related to the behavior of relational queries when viewed as mappings (as in [2]):

(iv) Given a database s and two queries Q_1, Q_2, testing whether $Q_1(s) \subseteq Q_2(s)$ and testing whether $Q_1(s)=Q_2(s)$ are both complete for the class Π_2^P of the polynomial-time hierarchy (see [11, 6] for relevant definitions).

(v) Given two databases s_1, s_2 and a query Q, testing whether $Q(s_1) \subseteq Q(s_2)$ and testing whether $Q(s_1)=Q(s_2)$ are both Π_2^P-complete.

In all the above results, databases can be constrained to consist of a single relation, and queries are restricted to only use the operations of projection and (natural) join.

2. Basic Definitions

The relational database model [3] assumes that the data are stored in tables called *relations.* The columns of a table correspond to *attributes*, and the rows to *tuples* (*records*). Each attribute A has an associated *domain* of values $Dom(A)$. A *relation scheme* is a finite set of attributes labeling the columns of a table, and it is usually written as a string of attributes. Let X be a relation scheme: an *X-tuple* is a mapping μ from X into $\bigcup_{A \in X} Dom(A)$ such that $\mu(A) \in Dom(A)$ for each attribute A in X; a *relation over* X is a finite set of X-tuples. A *database scheme* S is a finite set of relation schemes. A *database over* S is a set of relations containing exactly one relation over each relation scheme in S.

In the context of the relational model, one way of formulating queries is by using a set of operations defined on relations (relational algebra [3, 4]). In this paper we only consider two operations, *projection* and *join.*

The projection $t[Y]$ of an X-tuple t onto a subset Y of X is the restriction of t to Y. The projection $\pi_Y(R)$ of a relation R over X to Y is the set of projections of the tuples in R to Y. If R_1, R_2 are relations over the relation schemes X_1 and X_2 respectively, the join of R_1 and R_2, written $R_1 * R_2$, is the relation $R_1 * R_2 = \{\mu \mid \mu$ is an $X_1 \cup X_2$-tuple, $\mu[X_1] \in R_1$, $\mu[X_2] \in R_2\}$. The join of a finite set of relations $\{R_j\}$ will be written as $* \, R_j$.

A *relational expression* consists of relation schemes as operands and projection and join as operations. A relational expression φ defines, in the obvious way, a function which takes one argument for each relation scheme X appearing in the expression as operand (the corresponding argument is a relation over X), and produces as result a relation over a certain relation scheme, the *target relation scheme* of φ, $trs(\varphi)$. We will be using relational expressions to formulate queries in relational databases.

We also give the definition of the complexity class D^p: $D^p = \{L_1 \cap L_2 \colon L_1 \in NP,\ L_2 \in \text{co-}NP\}$. We have $NP \cup \text{co-}NP \subseteq D^p \subseteq \Delta_2{}^p = P^{NP}$ (polynomial time with an oracle from NP). We also note that, unless $NP = \text{co-}NP$, a problem which is complete for D^p is not in $NP \cup \text{co-}NP$. For a discussion of the type of languages of which D^p is the natural niche, and also for more natural problems which turn out to be complete for D^P, see [10].

3. D^P-completeness and $\#P$-hardness Results

We will first prove the following:

Theorem 1:

Given a relation R, a relational expression φ over projection and join, and a relation r, it is D^P-complete to test whether $\varphi(R)=r$.

Theorem 2:

Given a relation R, a relational expression φ over projection and join, and two "small" non-negative integers d_1, d_2 (i.e. written in unary), it is D^P-complete to test whether $d_1 \leq |\varphi(R)| \leq d_2$, even when $d_1 = d_2$ or when $d_1 < d_2$. Testing whether $d_1 \leq |\varphi(R)|$ is NP-complete, and testing whether $|\varphi(R)| \leq d_2$ is co-NP-complete.

All the reductions in this paper are reminiscent of the reduction in [1]. The reductions used to prove the D^P-completeness results in Theorem 1 and Theorem 2 are from the following problem:

3SAT-3UNSAT: "Given two Boolean expressions G, G' in 3-conjunctive normal form (3CNF), is it true that G is satisfiable and G' is not?"

It is immediate that 3SAT-3UNSAT is in D^P, and it is also a straightforward consequence of the fact that the 3-satisfiability problem is NP-complete [5, 6, 8] that 3SAT-3UNSAT is complete for D^P (see also [10]). Furthermore, it is clear that we may restrict ourselves to expressions containing at least three clauses, and such that the variables appearing in each clause are distinct.

Now let $G=F_1 \ldots F_m$ be a Boolean expression in 3-conjunctive normal form; the F_j's are clauses of three literals each and the variables appearing in the expression are $x_1, x_2, \ldots, x_n$. We denote the variables appearing in a clause F_j by $x_{j_1}, x_{j_2}, x_{j_3}$. We construct a relation R_G corresponding to G as follows: R_G has $n+1+m(m+1)/2$ columns. The first m columns correspond to the clauses of G and are labeled by the attributes $F_1, \ldots, F_m$; the next n columns correspond to the variables in G and are labeled by the attributes $X_1, X_2,...,X_n$; the next $m(m$-$1)/2$ columns correspond to the two-element subsets of $\{1,...,m\}$, and are labeled by the attributes $Y_{\{1,2\}},...,Y_{\{1,m\}},...,Y_{\{m-1,m\}}$; the last column is labeled by the attribute S.

For each clause F_j of G, R_G has 7 tuples as follows: let h_{jk}, $k=1,...,7$, be the seven satisfying truth assignments of the clause F_j (each h_{jk} is a function from $\{x_{j_1}, x_{j_2}, x_{j_3}\}$ to $\{0, 1\}$); for each h_{jk}, R_G contains a tuple μ_{jk} such that $\mu_{jk}(F_j)=1$, $\mu_{jk}(F_l)=e$ for $l\neq j$, $\mu_{jk}(X_{j_i})=h_{jk}(x_{j_i})$, $i=1,2,3$, $\mu_{jk}(X_l)=e$ for $l\neq j_i$, $i=1,2,3$, $\mu_{jk}(Y_{\{i,l\}})=x$ if $j=i$ or $j=l$ and $\mu_{jk}(Y_{\{i,l\}})=e$ otherwise, $\mu_{jk}(S)=a$. Finally, R_G contains a tuple ν, where $\nu(F_j)=1$, $\nu(S)=b$, $\nu(W)=e$ for $W\neq F_j$, S.

We also consider the following relational expression φ_G corresponding to G: $\varphi_G= \pi_{F_1...F_m}(T)\ *\ [\ *\ \ \pi_{F_jX_{j_1}X_{j_2}X_{j_3}Y_{\{j,1\}}...Y_{\{j,m\}}S}(T)\ ]$, where T is the relation scheme of R_G, namely $F_1...F_mX_1X_2...X_nY_{\{1,2\}}...Y_{\{1,m\}}...Y_{\{m-1,m\}}S$.

<u>*Example*</u>. Let $G=(x_1+x_2+x_3)(\neg x_2+x_3+\neg x_4)(\neg x_3+\neg x_4+\neg x_5)$ ($\neg x$ stands for the negation of the variable x); the relation R_G is

$F1$	$F2$	$F3$	$X1$	$X2$	$X3$	$X4$	$X5$	$Y\{1,2\}$	$Y\{1,3\}$	$Y\{2,3\}$	S
1	e	e	0	0	1	e	e	x	x	e	a
1	e	e	0	1	0	e	e	x	x	e	a
1	e	e	0	1	1	e	e	x	x	e	a
1	e	e	1	0	0	e	e	x	x	e	a
1	e	e	1	0	1	e	e	x	x	e	a
1	e	e	1	1	0	e	e	x	x	e	a
1	e	e	1	1	1	e	e	x	x	e	a
e	1	e	e	0	0	0	e	x	e	x	a
e	1	e	e	0	0	1	e	x	e	x	a
e	1	e	e	0	1	0	e	x	e	x	a
e	1	e	e	0	1	1	e	x	e	x	a
e	1	e	e	1	0	0	e	x	e	x	a
e	1	e	e	1	1	0	e	x	e	x	a
e	1	e	e	1	1	1	e	x	e	x	a
e	e	1	e	e	0	0	0	e	x	x	a
e	e	1	e	e	0	0	1	e	x	x	a
e	e	1	e	e	0	1	0	e	x	x	a
e	e	1	e	e	0	1	1	e	x	x	a
e	e	1	e	e	1	0	0	e	x	x	a
e	e	1	e	e	1	0	1	e	x	x	a
e	e	1	e	e	1	1	0	e	x	x	a
1	1	1	e	e	e	e	e	e	e	e	b

The relational expression φ_G is

$$\pi_{F_1F_2F_3} * \pi_{F_1X_1X_2X_3Y_{\{1,2\}}Y_{\{1,3\}}S} * \pi_{F_2X_2X_3X_4Y_{\{1,2\}}Y_{\{2,3\}}S} * \pi_{F_3X_3X_4X_5Y_{\{1,3\}}Y_{\{2,3\}}S}$$

We remark at this point that the fact that the same symbols (0, 1, e, x) are used in different columns is irrelevant; one could imagine replacing (in a consistent way, of course) any symbols in any particular column by new symbols, appearing only in that column.

R_G and φ_G can be constructed in time polynomial in the space needed to write down G; they capture the satisfiability (or unsatisfiability) of G as described below (let F denote the relation scheme $F_1...F_m$ and Y denote the relation scheme $Y_{\{1,2\}}...Y_{\{1,m\}}...Y_{\{m-1,m\}}$):

Lemma 1:

$\varphi_G(R_G)=R_G \cup R_\sigma$ where for each tuple μ in R_σ, $\mu(F_j)=1$, $\mu(Y_{\{i,l\}})=x$, $\mu(S)=a$, and $\mu[X_1X_2...X_n]$ defines a satisfying truth assignment h for G (by taking $h(x_i)=\mu(X_i)$); conversely, for each satisfying truth assignment for G there is a corresponding tuple in R_σ.

Proof: Let μ be a tuple in $\varphi_G(R_G)$. If $\mu(S)=b$, then $\mu=\nu$. If $\mu(S)=a$, then either $\mu[F]=\mu_{jk}[F]$ for some j, k, or $\mu[F]=\nu[F]$. In the first case, it is not difficult to check that $\mu=\mu_{jk}$ (this is enforced by the Y attributes); in the second case, it is clear that $\mu(F_j)=1$, $\mu(Y_{\{i,l\}})=x$, $\mu(S)=a$, and $\mu[X_1X_2...X_n]$ goes through exactly the satisfying truth assignments for G. ∎

Proposition 1:

If G is unsatisfiable, $\pi_Y\varphi_G(R_G)=\pi_Y(R_G)$; if G is satisfiable, $\pi_Y\varphi_G(R_G)=\pi_Y(R_G)\cup u_G$, where u_G is the Y-tuple such that $u_G(Y_{\{i,l\}})=x$.

Proof: Immediate, from Lemma 1. ∎

The following is a simple fact which we will also be using in subsequent proofs:

Proposition 2:

Given a relation R, a relational expression φ over projection and join, and a tuple t, testing whether $t \in \varphi(R)$ is in NP.

Proof: Immediate, by an inductive argument on the structure of φ. Alternatively, one may consider the *tableau* [1] corresponding to φ, and guess a valuation showing that $t\in\varphi(R)$. ∎

Notice that it immediately follows from Proposition 1 that G is satisfiable iff $u_G\in\pi_Y\varphi_G(R_G)$. Combining this fact with Proposition 2, we get a direct proof of the following result, proved indirectly in [16]:

"Given a relation R, a tuple t, and relation schemes X, Y_i, it is NP-complete to test whether $t \in \pi_X(* \pi_{Y_i}(R))$".

It also follows from Lemma 1 that G is unsatisfiable iff $\varphi_G(R_G)=R_G$, and combining with Proposition 2 we get a direct proof of the following result [9]:

"Given a relation R and relation schemes Y_i, testing whether $* \pi_{Y_i}(R)=R$ is co-NP-complete".

We now prove Theorem 1 and Theorem 2.

Proof of Theorem 1:

We first show membership in D^P: it suffices to show that testing whether $r\subseteq\varphi(R)$ is in NP, and testing whether $\varphi(R)\subseteq r$ is in co-NP. The first is a simple consequence of Proposition 2: just test whether $t\in\varphi(R)$ for all tuples $t\in r$. The second is equivalent to showing that testing whether $\varphi(R)\not\subseteq r$ is in NP ($\not\subseteq$ stands for the negation of $\subseteq$): for this, just nondeterministically guess a tuple t and check that $t\in\varphi(R)$, and also check that $t\notin r$.

For the D^P-hardness part, we will make a reduction from 3SAT-3UNSAT. Let G, G' be two Boolean expressions in 3CNF. Let R_G be the relation corresponding to G over the relation scheme $T=F_1...F_mX_1X_2...X_nY_{\{1,2\}}...Y_{\{1,m\}}...Y_{\{m-1,m\}}S$, and $R_{G'}$ be the relation corresponding to G' over the relation scheme $T'=F'_1...F'_{m'}X'_1X'_2...X'_{n'}Y'_{\{1,2\}}...Y'_{\{1,m'\}}...Y'_{\{m'-1,m'\}}S'$. Let $R_{G,G'}=R_G*R_{G'}$; observe that $T\cap T'=\emptyset$, and that for $Z\subseteq T$ we have $\pi_Z(R_{G,G'})=\pi_Z(R_G)$, and for $Z'\subseteq T'$ we have $\pi_{Z'}(R_{G,G'})=\pi_{Z'}(R_{G'})$. Let $\varphi_{G,G'}$ be the relational expression $\pi_{YY'}(\varphi_G*\varphi_{G'})$, taking as argument the relation scheme $T\cup T'$; clearly $\varphi_{G,G'}(R_{G,G'})=\pi_Y\varphi_G(R_G)$

$* \; \pi_{Y'}\varphi_{G'}(R_{G'})$, and thus using Proposition 1 it is easy to see that G is satisfiable and G' is unsatisfiable iff $\varphi_{G,G'}(R_{G,G'})=(\pi_Y(R_G)\cup u_G) * \pi_{Y'}(R_{G'})$ (call this relation $r_{G,G'}$). Since $R_{G,G'}$, $\varphi_{G,G'}$, $r_{G,G'}$ can be constructed in time polynomial in the space needed to write down G, G', we are done . ∎

Observe that in order to be able to make the reduction (and so in order for Theorem 1 to be true) it suffices to consider relational expressions of a certain restricted form. This will be the case for all of our results.

Proof of Theorem 2:

To prove the membership assertions, it suffices to show that testing whether $d_1 \leq |\varphi(R)|$ is in NP and testing whether $|\varphi(R)| \leq d_2$ is in co-NP. The first follows from Proposition 2; just guess nondeterministically d_1 distinct tuples (recall that d_1 is in unary) and check that each of them is in $\varphi(R)$. For the second it suffices to show that testing whether $|\varphi(R)| \geq d_2+1$ is in NP, which is the same as the first.

For the D^P-hardness part let G, G' be as before, and let $\beta=7m+1$, $\beta'=7m'+1$. By appropriately padding G' (add a sufficient number of extra clauses not affecting its satisfiability) we can make sure that $\beta<\beta'$. Now $|\varphi_{G,G'}(R_{G,G'})|=|\pi_Y\varphi_G(R_G)|\;|\pi_{Y'}\varphi_{G'}(R_{G'})|$, and furthermore $|\pi_Y\varphi_G(R_G)|=\beta$ if G is unsatisfiable and $|\pi_Y\varphi_G(R_G)|=\beta+1$ if G is satisfiable, and similarly for G' (Proposition 1). Thus, G is satisfiable and G' is unsatisfiable iff $|\varphi_{G,G'}(R_{G,G'})|=(\beta+1)\beta'$, iff $\beta(\beta'+1)+1 \leq |\varphi_{G,G'}(R_{G,G'})| \leq \beta(\beta'+1)+\beta'$.

For the remaining hardness assertions it suffices to observe that, by Lemma 1, G is satisfiable iff $\beta+1 \leq |\varphi_G(R_G)|$, and G is unsatisfiable iff $|\varphi_G(R_G)| \leq \beta$. ∎

Using the constructions described, we can also show that counting the number of tuples in $Q(s)$, where s is a database and Q a relational query, is at least as hard as counting the number of accepting computations of any polynomial time nondeterministic Turing machine. We make use of the fact that the problem of counting the number of satisfying truth assignments for an instance of 3-SATISFIABILITY is $\#P$-complete [14].

Theorem 3:

Given a relation R and a relational expression φ over projection and join, the enumeration problem of counting the number of tuples in $\varphi(R)$ is $\#P$-hard.

Proof: Immediate, by the fact that if $a(G)$ denotes the number of satisfying truth assignments for G, then $a(G)=|\varphi_G(R_G)|-7m-1$ (Lemma 1). ∎

By observing the form of φ_G we can also see the following:

Corollary:

Given a relation R and relation schemes Y_i, the enumeration problem of counting the number of tuples in $*\ \pi_{Y_i}(R)$ is $\#P$-complete.

Proof: By the proof of Theorem 3, it suffices to show membership in $\#P$. Consider the following "counting Turing machine" M: M nondeterministically guesses a tuple t, and accepts if and only if $t[Y_i]\in R$ for all i. Clearly M runs in polynomial time, and the number of accepting computations of M is the same as the number of tuples in $*\ \pi_{Y_i}(R)$. ∎

4. Π_2^P-completeness Results

In this section we prove the following:

Theorem 4:

Given a relation R and two relational expressions φ_1, φ_2 over projection and join, testing whether $\varphi_1(R)\subseteq\varphi_2(R)$ and testing whether $\varphi_1(R)=\varphi_2(R)$ are both Π_2^P-complete.

Theorem 5:

Given two relations R_1, R_2 and a relational expression φ over projection and join, testing whether $\varphi(R_1)\subseteq\varphi(R_2)$ and testing whether $\varphi(R_1)=\varphi(R_2)$ are both Π_2^P-complete.

We first prove that the problems above are in Π_2^P.

Proposition 3: Given two relations R_1, R_2, and two relational expressions φ_1, φ_2 over projection and join, testing whether $\varphi_1(R_1) \subseteq \varphi_2(R_2)$ and testing whether $\varphi_1(R_1) = \varphi_2(R_2)$ are both in Π_2^p.

Proof: For the first part, it suffices to show that testing whether $\varphi_1(R_1) \not\subseteq \varphi_2(R_2)$ is in $\Sigma_2^p = NP^{NP}$ (nondeterministic polynomial time with an oracle from *NP*). But by Proposition 2 all we have to do is nondeterministically guess a tuple t and check, by asking an appropriate oracle from *NP*, that $t \in \varphi_1(R_1)$ and $t \notin \varphi_2(R_2)$. The second part is similar. ∎

The reductions used to prove Theorem 5 and Theorem 6 are from the following problem, which was shown to be Π_2^p-complete in [11, 15]:

Q-3SAT: "Given a Boolean expression G in 3-conjunctive normal form and a partition of the variables in G into two sets $X = \{x_1, ..., x_r\}$, $X' = \{x_{r+1}, ..., x_n\}$, determine whether for all assignments of truth values to the variables in X G is satisfiable, i.e. determine whether $\forall X \exists X'(G(X, X') = 1)$ (we write $\forall X$ for $\forall x_1 ... \forall x_r$ etc.)".

We will actually need to impose the following technical restrictions on the set X:

Proposition 4:

Q-3SAT is Π_2^p-complete even if X is not contained in any V_j, where V_j is the set of variables appearing in the clause F_j, and also X does not contain any V_j.

Proof: To enforce the first restriction, just add to G the clauses $(v_1 + v_2 + v_3)$, $(v_4 + v_5 + v_6)$, where the v_i's are new variables, and replace X by $X \cup \{v_1, v_4\}$. If X contains V_j then the problem is trivial, because there is an assignment to the variables in V_j making F_j false, and thus $\forall X \exists X'(G(X, X') = 1)$ is false. ∎

In the following reductions, we will assume that the restrictions described above are satisfied. We will use X to denote both the set $\{x_1, ..., x_r\}$ and the relation scheme $X_1 ... X_r$

Proof of Theorem 4:

By Proposition 3, we only need to prove the hardness part. Let G, X be as above, and let R'_G be a relation obtained from R_G as follows: first, for each clause F_j add a tuple ξ_j corresponding to the truth assignment h_j: $\{x_{j_1}, x_{j_2}, x_{j_3}\} \rightarrow \{0, 1\}$ which does not satisfy F_j; that is, $\xi_j(F_j)=1$, $\xi_j(F_l)=e$ for $l\neq j$, $\xi_j(X_{j_i})=h_j(x_{j_i})$, $i=1,2,3$, $\xi_j(X_l)=e$ for $l\neq j_i$, $i=1,2,3$, $\xi_j(Y_{\{i,l\}})=x$ if $j=i$ or $j=l$ and $\xi_j(Y_{\{i,l\}})=e$ otherwise, $\xi_j(S)=a$. Then add a new column labeled by U, and make $\xi_j(U)=c_j$, and $\mu(U)=c$ for any other tuple μ in the relation.

Consider also the following relational expressions (T' is the relation schemeof R'_G, namely $F_1...F_mX_1X_2...X_nY_{\{1,2\}}...Y_{\{1,m\}}...Y_{\{m-1,m\}}SU$):

$$\varphi^1{}_G=\pi_{F_1...F_m}(T') \;*\; [\,*\; \pi_{F_jX_{j_1}X_{j_2}X_{j_3}Y_{\{j,1\}}...Y_{\{j,m\}}S}(T')\,] \;*\; \pi_U(T')$$

$$\varphi^2{}_G=\pi_{F_1...F_m}(T') \;*\; [\,*\; \pi_{F_jX_{j_1}X_{j_2}X_{j_3}Y_{\{j,1\}}...Y_{\{j,m\}}SU}(T')\,]$$

It is not difficult to see that $\varphi^2{}_G$ picks out the satisfying truth assignments for G (despite the extra tuples, since it looks at both the S and U columns), while $\varphi^1{}_G$ considers G as a tautology (because of the extra tuples, since it only looks at the S column). More specifically, we have that $\pi_X\varphi^1{}_G(R'_G)=\pi_X(R'_G)\cup R_X$ and $\pi_X\varphi^2{}_G(R'_G)=\pi_X(R'_G)\cup R_{X,a}$, where:

a) R_X consists of all possible truth assignments to variables in X, and $R_{X,a}$ consists of the restrictions of the satisfying truth assignments for G to the variables in X ($R_{X,a}=\pi_X(R_a)$).

b) No tuple in $\pi_X(R'_G)$ may be taken as defining a truth assignment (by our first restriction on X, any such tuple contains at least one e).

Thus, $\forall X\exists X'(G(X, X')=1)$ is true iff $R_X\subseteq R_{X,a}$, iff $\pi_X(R'_G)\cup R_X\subseteq\pi_X(R'_G)\cup R_{X,a}$, iff $\pi_X\varphi^1{}_G(R'_G)\subseteq\pi_X\varphi^2{}_G(R'_G)$, iff $\pi_X\varphi^1{}_G(R'_G)=\pi_X\varphi^2{}_G(R'_G)$. This completes the proof. ∎

Proof of Theorem 5:

By Proposition 3, we only have to prove the hardness part. Let G, X be as above, and let R''_G be a relation obtained from R'_G by omitting the U column. We now have $\pi_X\varphi_G(R''_G)=\pi_X(R''_G)\cup R_X$ and $\pi_X\varphi_G(R_G)=\pi_X(R_G)\cup R_{X,a}$. As before, no tuple in either $\pi_X(R''_G)$ or $\pi_X(R_G)$ may be taken as defining a truth assignment; moreover, we actually have $\pi_X(R''_G)=\pi_X(R_G)$ (by our second restriction on X, the extra tuples in R''_G do not matter).Thus, $\forall X\exists X'(G(X, X')=1)$ is true iff $R_X\subseteq R_{X,a}$, iff $\pi_X(R''_G)\cup R_X\subseteq\pi_X(R_G)\cup R_{X,a}$, iff $\pi_X\varphi_G(R''_G)\subseteq\pi_X\varphi_G(R_G)$, iff $\pi_X\varphi_G(R''_G)=\pi_X\varphi_G(R_G)$. This completes the proof. ∎

Observe also that the following fact can be proved by a trivial modification to our proofs:

"Given a relation R, a relational expression φ over projection and join, and for each attribute X_i in $trs(\varphi)$, a finite subset E_i of $Dom(X_i)$, it is Π_2^p-complete to test whether $\varphi(R)$ contains every possible tuple μ such that $\mu(X_i) \in E_i$ for all X_i in $trs(\varphi)$".

5. Discussion

The results of Section 3 must be seen as further theoretical evidence supporting the intuitive fact that processing a query requires in general time exponential in the size of the query. Moreover, they imply that the intermediate results can be inherently much larger than both the input relation and the query output They also provide strong evidence of the intractability of the problem of estimating the size of the query output.

The results of Section 4 characterize the complexity of testing the equivalence of relational queries on a fixed given relation (Theorem 4). They are also motivated by the theory of database mappings, which was proposed as a means to solve the view update problem [2]. If S is a general state space of possible values of a database and f is a mapping on S, one considers the partition of S induced by the equivalence relation $\sim_f$ where $x \sim_f y$ iff $f(x)=f(y)$. Theorem 5 indicates the hardness of deciding this condition in the special case of relational mappings.

Acknowledgement

The author wishes to thank Professor Christos Papadimitriou and Mihalis Yannakakis for many helpful suggestions and comments.

References

(Note: References [7, 12] are not cited in the text.)

[1] Aho, A.V., Sagiv, Y., and Ullman, J.D. Equivalences among relational expressions. SIAM Journal of Computing 8, 2 (May 1979), 218-246.

[2] Bancilhon, F.M., Spyratos, N. Data Base Mappings, Part I: Theory. Rapport de Recherche INRIA No. 62.

[3] Codd, E.F. A relational model for large shared data banks. Communications of the ACM 13, 6 (June 1970), 377-387.

[4] Codd, E.F. Relational completeness of data base sublanguages. In Data Base Systems, R.Rustin, Ed., Prentice Hall, Englewood Cliffs, N.J., 1972, pp. 65-98.

[5] Cook, S.A. The complexity of theorem proving procedures. Proceedings of the 3rd Annual ACM Symposium on the Theory of Computing, Shaker Heights, Ohio, May 1971, pp. 151-158.

[6] Garey, M.R. and Johnson, D.S. Computers and Intractability: A Guide to the Theory of NP-completeness (Freeman, San Francisco, CA, 1979).

[7] Honeyman, P., Ladner, R., Yannakakis, M. Testing the universal instance assumption. Information Processing Letters 10, 1, 14-19 (1980).

[8] Karp, R.M. Reducibility among combinatorial problems. In Complexity of Computer Computations, R.E. Miller and J.W. Thatcher, Eds., Plenum Press, New York, 1972, pp. 85-104.

[9] Maier, D., Sagiv, Y., and Yannakakis, M. On the complexity of testing implications of functional and join dependencies. Journal of the ACM 28, 4, 680-695 (1981).

[10] Papadimitriou, C.H., Yannakakis, M. The complexity of facets (and some facets of complexity). Proceedings of the 14th Annual ACM Symposium on the Theory of Computing, San Francisco, California, May 1982, pp. 255-260.

[11] Stockmeyer, L.J. The polynomial time hierarchy. Theoretical Computer Science 3, 1 (1976), 1-22.

[12] Sagiv, Y., and Yannakakis, M. Equivalences among relational expressions with the union and difference operators. Journal of the ACM 27, 4, 633-655 (1980).

[13] Valiant, L.G. The complexity of computing the permanent. Report No. CSR-14-77, Computer Science Department, University of Edinburgh, Edinburgh, Scotland, 1977.

[14] Valiant, L.G. The complexity of enumeration and reliability problems. SIAM Journal of Computing 8, 3 (August 1979), 410-421.

[15] Wrathall, C. Complete sets and the polynomial-time hierarchy. Theoretical Computer Science 3, 1, (1976), 23-33.

[16] Yannakakis, M. Algorithms for acyclic database schemes. Proceedings of the 7th VLDB Conference, 1981, pp. 82-94.