On the Π_2^P-Completeness of the Containment Problem of Conjunctive Queries with Negation and Other Problems

Marie-Laure Mugnier

LIRMM, Université de Montpellier,
161, rue Ada, F-34392 Montpellier cedex - France
mugnier@lirmm.fr

Abstract. This research note proves the Π_2^P-completeness of several equivalent problems: deduction in the first-order logical fragment composed of existential conjunctive formulas, deduction in polarized graphs (which are simple conceptual graphs with negation on relations, built on a simplified vocabulary) and containment of conjunctive queries with negation. The exhibited reduction is essentially due to Guillaume Bagan (November 2004)[1].

Research Report LIRMM number 07004 - February 2007

1 Some equivalent problems

We consider the three following equivalent problems:

Deduction in FOL$\{\exists, \wedge, \neg_a\}$
Input: two existentially closed conjunctions of (positive and negative) literals f_1 and f_2.
Question: does $f_1 \models f_2$ hold, i.e. is every model of f_1 a model of f_2?

To ease comparison with the following problems, we consider that the formulas are in prenex form (all quantifiers are in front of the formula).

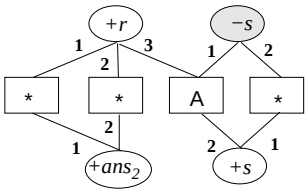
Deduction on polarized graphs (PGs)
Input: two polarized graphs G_1 and G_2.
Question: is G_2 deducible from G_1, i.e. is G_1 inconsistent or is there a homomorphism from G_2 to each completion of G_1?

For the definitions of polarized graph, homomorphism and completion, see section 3. See also [LM07] in relationship with the containment problem of conjunctive queries with negation.

[1] Guillaume Bagan proposed this reduction when he was a master student in Montpellier University in November 2004. It was an answer to the question of the complexity of deduction on polarized graphs, that I raised during a master course.

Containment of conjunctive queries with negation
Input: two conjunctive queries with negation (but without inequalities) q_1 and q_2.
Question: is q_1 contained in q_2 ($q_1 \sqsubseteq q_2$), i.e. is the set of answers to q_1 included in the set of answers to q_2 for any database D?

Polarized graphs can be seen as a graph representation of FOL$\{\exists, \wedge, \neg_a\}$ formulas. More specifically, the logical translation of a polarized graph is a FOL$\{\exists, \wedge, \neg_a\}$ formula (see the ϕ mapping, section 3), and each FOL$\{\exists, \wedge, \neg_a\}$ formula can be represented as a polarized graph, such that the notions of deduction coincide[2]. The equivalence between the two first problems is shown in [ML06] (on more complex polarized graphs). The logical view of queries has long been used in databases. See for instance [AHV95]. Figure 1 illustrates these different views of the same object.



f=$\exists x \exists y \exists z(ans(x, y) \wedge r(x, y, A) \wedge s(z, A) \wedge \neg s(A, z))$
q=$ans(x, y) \leftarrow r(x, y, A) \wedge s(z, A) \wedge \neg s(A, z))$

Fig. 1. Equivalent views of the same object: PG, FOL$\{\exists, \wedge, \neg_a\}$ formula and conjunctive query with negation

One can identify the notions of substitution on FOL$\{\exists, \wedge, \neg_a\}$ formulas (in prenex form), PG homomorphism and query homomorphism extended to the negative literals (or subgoals) of a query. Similarly, one can identify the notions of canonical[3] model of a FOL$\{\exists, \wedge, \neg_a\}$ formula, completion of a consistent PG and completion of a consistent conjunctive query with negation.

2 Π_2^P-completeness Proof

The way the second problem is stated makes it clear that it is in Π_2^P. Note that a completion has a size exponential in the maximum arity of a relation, but without loss of generality we can assume that this size is bounded by a constant (or even by 2, as n-ary relations can be polynomially transformed into binary relations).

The Π_2^P-completeness is proven by reduction from the following Π_2^P-complete problem:

[2] If we restrict the domain of ϕ to *normal* polarized graphs, i.e. PGs without two vertices labeled by the same constant (the PG in Figure 1 is normal), ϕ is a bijection.
[3] In a canonical model, the domain is the set of terms occurring in the formula.

Generalized Ramsey Number
Input: an undirected graph $G = (V, E)$, where V is the set of vertices and E the set of edges; a partial two-coloring σ of E (i.e. a partial function $\sigma : E \mapsto \{0, 1\}$); and an integer k.
Question: Does every total extension of σ produce a one-color k-clique[4] (i.e. a k-clique with all edges colored 0 or with all edges colored 1)?

See [SU02] for references to the Π_2^P-completeness proof.

Reduction. Let $(G = (V, E), \sigma, k)$ be an instance of Generalized Ramsey Number. Let $L = \{r, d, p\}$ be a relational vocabulary where r and d are binary relations and p is a unary relation. We build two relational structures G_1 and G_2 on L which can naturally be seen as (consistent) polarized graphs (with all term vertices labeled by $*$). Hence (G_1, G_2) is an instance of PG-deduction. G_1 and G_2 can also be seen as FOL$\{\exists, \wedge, \neg_a\}$ formulas or conjunctive queries with negation. A term vertex (resp. relation vertex) in a polarized graph corresponds to a term (resp. literal) in the associated formula or query; $r(u)$ (resp. $\neg r(u)$) denotes a positive (resp. negative) relation vertex with label $+r$ (resp. $-r$) and list of neighbors u. Therefore, the following reduction is easily translated into a reduction to the other problems.

The set of term vertices of G_2 is $C_1 \cup C_2$ where C_1 and C_2 are copies of the set of edges of a k-clique (thus in each C_i there is one term vertex x for each edge ab of a k-clique). There is a $r(x, y)$ if $\{x, y\} \subseteq C_1$ or $\{x, y\} \subseteq C_2$, and x and y represent distinct edges sharing a vertex (i.e. edges with a common endpoint) in the k-clique. There is a $d(x, y)$ between all distinct term vertices x and y in G_2. There is a $p(x)$ if $x \in C_1$ and $\neg p(x)$ if $x \in C_2$.

Intuitively: G_2 is composed of two components, C_1 and C_2; for each component, the term vertices and the relation vertices labeled by r represent the intersection graph of the edges of a k-clique; distinct term vertices anywhere in G_2 are related by d; the term vertices in C_1 are marked by p, those in C_2 are marked by $\neg p$. p can be seen as denoting the color 0 and $\neg p$ the color 1.

The set of term vertices of G_1 is $C_1' \cup E \cup C_2'$ where, as for G_2, C_1' and C_2' are copies of the set of edges of a k-clique; E is a copy of the set of edges in G.

There is a $r(x, y)$ if $\{x, y\} \subseteq C_1'$ or $\{x, y\} \subseteq C_2'$ or $\{x, y\} \subseteq E$, and x and y represent distinct edges sharing a vertex. There is a $d(x, y)$ between distinct vertices x and y in $C_1' \cup E$ and between distinct vertices x and y in $E \cup C_2'$[5]. There is a $p(x)$ if $x \in C_1'$, or $x \in E$ and $\sigma(x)$ is defined and equal to 0; there is a $\neg p(x)$ if $x \in C_2'$, or $x \in E$ and $\sigma(x)$ is defined and equal to 1.

Intuitively: G_1 has the same components C_1 and C_2 as G_2, but in addition there is a component E representing the intersection graph of the edges of G. The elements in E are marked by p, $\neg p$ or nothing, depending of the color of the corresponding edge in G. In G_2, C_1 and C_2 are "directly connected" by d; in G_1 they are not: E is between them.

[4] A clique is an undirected graph with all edges between distinct vertices. A k-clique is a clique with k vertices.

[5] It is not necessary to have d in both directions between C_1 and C_2 in G_2, and between C_1' and E and E and $C'2$ in G_1: instead, d could lead from C_1 to C_2 in G_2, and from C_1' to E and from E to $C'2$ in G_1.

Now, let us prove that all extension of σ produces a one-color k-clique if and only if for all completion $G*_1$ of G_1, there is a homomorphism from G_2 to $G*_1$.

Note. We call subgraph "induced by" a set of term vertices S the subgraph with set of vertices S plus all relation vertices having their neighbors in S.

First see that there is a *bijection* (say b) between an edge-coloring of G extending σ and a $\{p\}$-completion of the subgraph induced by E in G_1 . As the subgraphs induced by C'_1 and C'_2 are already $\{p\}$-complete, such a completion is also a $\{p\}$-completion of G_1 itself. Also note that r and d occur only positively, thus they are not needed in completions: there is a homomorphism from G_2 to each $\{r, d, p\}$-completion of G_1 iff there is a homomorphism from G_2 to each $\{p\}$-completion of G_1 (see property 2, section 3).

Let σ' be an extension of σ which produces a k-clique of color 0 (resp. 1). In $b(\sigma')$, i.e. the completion of G_1 assigned to σ' by b, the subgraph induced by E contains term vertices corresponding to the edges of a k-clique S, with all these vertices being attached to a p relation vertex (resp. a $\neg p$ relation vertex). Let us note $edges(S)$ this set of vertices. There is a homomorphism from G_2 to the subgraph of $b(\sigma')$ induced by $C'_1 \cup edges(S)$ (resp. $C'_2 \cup edges(S)$).

Reciprocally, let $G*_1$ be a $\{p\}$-completion of G_1 and let h be a homomorphism from G_2 to $G*_1$. See that h is necessarily injective on the set of term vertices in G_2 due to the p relation vertices between all distinct term vertices in G_2[6].h maps G_2 either to the subgraph of $G*_1$ induced by C'_1 and the vertices corresponding to the edges of a k-clique all attached to a $\neg p$, or to the subgraph induced by C'_2 and the edges of a k-clique all attached to a p. Thus, there is an extension of σ which produces a k-clique of color 0 or of color 1.

3 Definitions and results relative to polarized graphs

The *polarized graphs* presented below are a simplification of graphs used in a knowledge representation context [Ker01,ML06], where the vocabulary is more complex: relation names are partially ordered; moreover term vertices are typed, with the types being taken in a partially ordered set of types. See [ML06] for further details.

Definition 1 (polarized graph). *Let us consider a vocabulary $(\mathcal{R}, dom)$ where $\mathcal{R}$ is a finite set of relation names of any arity and dom a set of constant labels. A* polarized graph *(PG) is an undirected bipartite graph $G = (R,\ T,\ E,\ l)$ where R and T are the (disjoint) sets of vertices, respectively called set of* relation *vertices and set of* term *vertices, E is the family of edges (there may be several edges with the same extremities) and l is the label mapping. For $r_i \in R$, $l(r_i) = +r$ (r_i is called a positive relation vertex) or $l(r_i) = -r$ (r_i is called a negative relation vertex) where $r \in \mathcal{R}$; the degree of r_i (i.e. the number of edges incident to it) must be equal to the arity of r ; furthermore, the edges incident to r_i are totally ordered, which is represented by labeling edges from*

[6] A homomorphism from a clique C to a graph G is necessarily injective, but it is not true anymore when we replace C and G by the intersection graphs of their edges. The role of the d relation vertices in the subgraphs induced by C_1 and by C_2 is to enforce injectivity.

1 to the degree of r_i. An edge labeled n between a relation vertex r_i and a term vertex t is denoted by $(r_i,\ n,\ t)$. For $t \in T$, either $l(t) = *$ (t is called a variable node) or $l(t) \in dom$ (t is called a constant node).

We consider here, without loss of generality, that PGs do not have redundant relation vertices (i.e. with the same label and the same $i\text{-}ith$ neighbors) and that each constant of dom appears at most once in it.

Definition 2 (PG homomorphism). *A PG homomorphism h from $G = (R_G,\ T_G,\ E_G,\ l_G)$ to $H = (R_H,\ T_H,\ E_H,\ l_H)$ is a mapping from $R_G \cup T_G$ to $R_H \cup T_H$ such that:*

1. *for all $r \in R_G$, $h(r) \in R_H$; for all $t \in T_G$, $h(t) \in T_H$*
   (h preserves bipartition*)*
2. *if $(r,\ n,\ t) \in G$ then $(h(r),\ n,\ h(t)) \in H$*
   (h preserves edges and their ordering*)*
3. *for all $r \in R_G$, $l_H(h(r)) = l_G(r)$*
   (h preserves relation labels*)*
4. *for all $t \in T_G$, if $l_G(t) \in dom$ then $l_H(h(t)) = l_G(t)$, otherwise there is no condition on $l_H(h(t))$*
   (h may "instantiate" variables*).*

Definition 3 (inconsistent PG). *A PG is said to be* inconsistent *if it contains two relation nodes with labels $(+r)$ and $(-r)$ and same neighborhood. Otherwise it is said to be* consistent*.*

The semantics of PGs is given by a translation to first-order-logic, say ϕ (according to the name given to this mapping in conceptual graphs): briefly said, term nodes are translated into terms (a distinct variable is assigned to each variable node) and relation nodes into literals; the obtained formula is the existential closure of the conjunction of all literals. According to this semantic, we often note r and $\neg r$ the relation node labels instead of $+r$ and $-r$.

It can be easily checked that inconsistent PGs correspond to unsatisfiable formulas. Positive PGs are translated into positive formulas; for this positive fragment it has been proven that PG homomorphism is equivalent to logical deduction ([CM92], considering that positive PGs are a particular case of simple conceptual graphs).

Definition 4. *A consistent PG G is* complete *w.r.t. a set of relations R (in short: R-complete), if for each relation r in R with arity k, for each k-tuple of term nodes u in G, not necessarily distinct, G contains $r(u)$ (a positive relation node labeled $+r$ with list of neighbors u) or $\neg r(u)$ (a negative relation node labeled $-r$ with list of neighbors u).*

A complete PG is obtained from a PG G by repeatedly adding positive and negative relation nodes (with neighbors already present in G), as long as it does not yield a redundancy or an inconsistency. The following property justifies the definition of the "PG deduction" problem given in the first section.

Property 1. [ML06] Given two PGs G_1 and G_2 (with G_1 being consistent), $\phi(G_1) \models \phi(G_2)$ iff for each complete PG G_1^c generated from G_1 (w.r.t. the set of relation names occurring in G_1), there is a homomorphism from G_2 to G_1^c.

Not all relation names occurring in G_1 need to be considered:

Property 2. [LM07] If r is a relation name that does not have both positive and negative occurrences in G_1 and in G_2, then r is not needed in the completion of G_1 (i.e. G_2 is deducible from G_1 iff G_2 can be mapped by a homomorphism to each completion of G_1 built without considering r).

4 PGs and conjunctive queries with negation

Let us first recall classical database definitions (see f.i. [AHV95]). A *database schema* $S = (R, dom)$ includes a finite set of relations R and a countably infinite set of constants dom. Each relation has an *arity* (not equal to zero) defining the number of its arguments. A *database instance* D (or simply a *database*) over S maps each *k*-ary relation r_i of R to a finite subset of dom^k (denoted $D(r_i)$). A *conjunctive query (with negation)* is of form:

$$q = ans(u) \leftarrow r_1(u_1),\ ...\ r_n(u_n), \neg s_1(v_1),\ ...\ \neg s_m(v_m)\ \ n \geq 0,\ m \geq 0,\ n + m \geq 1$$

where $r_1\ ...\ r_n, s_1\ ...\ s_m$ are relations, ans is a special relation not belonging to R, u and $u_1\ ...\ u_n, v_1\ ...\ v_m$ are tuples of terms (variables or constants of dom), and each variable of u occurs at least once in the body of the rule. Without loss of generality, we assume that the same literal does not appear twice in the body of the rule. A *positive query* is a query without negative literals ($m = 0$, thus $n \geq 1$). A query is *inconsistent* if it contains two opposite literals (i.e. $\exists\ i, j\ 1 \leq i \leq n,\ 1 \leq j \leq m$ such that $r_i(u_i) = s_j(v_j)$), otherwise it is consistent.

Given a query $q = ans(u) \leftarrow r_1(u_1),\ ...\ r_n(u_n),\ \neg s_1(v_1),\ ...\ \neg s_m(v_m)$ and a database D on S, $q(D)$ denotes the *set of answers* to q in D; $q(D)$ is the set of tuples $\mu(u)$ where μ is a substitution of the variables in q by constants in dom such that for any i in $\{1, ..., n\}$, $\mu(u_i) \in D(r_i)$ and for any j in $\{1, ..., m\}$, $\mu(v_j) \notin D(s_j)$. When the arity of ans is 0, $q(D)$ is the set $\{()\}$ if there is such a substitution μ, otherwise it is $\emptyset$. If $q(D)$ is not empty, D is said to *answer* the query. A query q_1 is said to be *contained* in a query q_2, noted $q_1 \sqsubseteq q_2$, if for any database D, $q_1(D) \subseteq q_2(D)$. The *conjunctive query containment problem* takes as input two conjunctive queries q_1 and q_2 and asks whether $q_1 \sqsubseteq q_2$.

There is an obvious mapping from databases notions to PG notions. To a database schema is naturally assigned a vocabulary of PGs. A database D can naturally be seen as a totally instantiated (i.e. without variable vertices) positive PG on this vocabulary. The body q_b of a query q can naturally be seen as a PG. A whole query q can be transformed into a PG as follows. We add to the vocabulary special relation names ans_i for each possible arity i, possibly 0 (which corresponds to boolean queries). Then the head of q (say $ans(t_1...t_q)$) is mapped to a positive relation node with label ans_q and with neighborhood the term nodes assigned to t_i [7]. A query homomorphism extended to negative literals can then be seen as a PG homomorphism.

[7] This transformation is from [CMS98]. Note that we could use one unary relation per position in the head as in the transformation from a query to a database in [CM77] but then only queries with the same arity of ans should be compared.

References

[AHV95] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[CM77] A.K. Chandra and P.M. Merlin. Optimal implementation of conjunctive queries in relational databases. In *9th ACM Symposium on Theory of Computing*, pages 77–90, 1977.

[CM92] M. Chein and M.-L. Mugnier. Conceptual Graphs: Fundamental Notions. *Revue d'Intelligence Artificielle*, 6(4):365–406, 1992.

[CMS98] M. Chein, M.-L. Mugnier, and G. Simonet. Nested Graphs: A Graph-based Knowledge Representation Model with FOL Semantics. In *Proc. of KR'98*, pages 524–534. Morgan Kaufmann, 1998. Revised version available at http://www.lirmm.fr/~mugnier/.

[Ker01] G. Kerdiles. *Saying it with Pictures: a Logical Landscape of Conceptual Graphs*. PhD thesis, Univ. Montpellier II / Amsterdam, Nov. 2001.

[LM07] M. Leclère and M.L. Mugnier. Some algorithmic improvments for the containment problem of conjunctive queries with negation. In *ICDT'07*, number 4353 in LNCS, pages 404–418. Springer, 2007.

[ML06] M.L. Mugnier and M. Leclère. On querying simple conceptual graphs with negation. *Data and Knowledge Engineering (DKE)*, 2006. In press, doi:10.1016/j.datak.2006.03.008. To be published in Volume 60, Issue 3, Pages 435-624 (March 2007).

[SU02] Marcus Schaefer and Chris Umans. Completeness in the polynomial-time hierarchy: A compendium. Sigact News. Last updated: 8/20/06, available on M. Schaefer's homepagee, 2002.