Attacking Diophantus: Solving a Special Case of Bag Containment

George Konstantinidis
g.konstantinidis@soton.ac.uk
University of Southampton

Fabio Mogavero
fabio.mogavero@unina.it
Università degli Studi di Napoli Federico II

ABSTRACT

Conjunctive-query containment is the problem of deciding whether the answers of a given conjunctive query on an arbitrary database instance are always contained in the answers of a second query on the same instance. This is a very relevant question in query optimization, data integration, and other data management and artificial intelligence areas. The problem has been deeply studied and understood for the, so-called, *set-semantics*, *i.e.*, when query answers and database instances are modelled as sets of tuples. In particular, it has been shown by Chandra and Merlin to be NPTime-complete. On the contrary, when investigated under *bag-semantics*, *a.k.a. multiset semantics*, which allows for replicated tuples both in the underlying instance and in the query answers, it is not even clear whether the problem is decidable. Since this is exactly the standard interpretation for commercial relational database systems, the question turns out to be an important one. Multiple works on variations and restrictions of the *bag-containment problem* have been reported in the literature and, although the general problem is still open, we contribute with this article by solving a special case that has been identified as a major open problem on its own. More specifically, we study *projection-free queries*, *i.e.*, queries without existentially quantified variables, and show decidability for the bag-containment problem of a projection-free conjunctive query into a generic conjunctive query. We prove indeed that deciding containment in this setting is in Π_2^{p}. Our approach relies on the solution of a special case of the *Diophantine inequality problem* via a reduction to the *linear inequality problem* and clearly exposes inherent difficulties in the analysis of the general question.

PODS'19, June 30-July 5, 2019, Amsterdam, Netherlands

ACM ISBN 978-1-4503-6227-6/19/06…$15.00
https://doi.org/10.1145/3294052.3319689

KEYWORDS

Query Containment; Bag Semantics; Bag Containment; Multiset Semantics; Diophantine Inequalities

ACM Reference Format:
George Konstantinidis and Fabio Mogavero. 2019. Attacking Diophantus: Solving a Special Case of Bag Containment. In *38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS'19), June 30-July 5, 2019, Amsterdam, Netherlands.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3294052.3319689

1 INTRODUCTION

The standard *query containment problem* can be defined as follows: given a *database schema* $\mathcal{S}$ and two *queries* q_1 and q_2 on $\mathcal{S}$, decide whether, for all *relational database instances* [12] $\mathcal{I}$ over $\mathcal{S}$, it is the case that $q_1^{\mathcal{I}} \subseteq q_2^{\mathcal{I}}$, in symbols $q_1 \sqsubseteq q_2$, with $q_i^{\mathcal{I}}$ being the set of answer tuples of query q_i on database $\mathcal{I}$. This is a problem of both great theoretical and practical significance, fundamental to databases [1] and knowledge representation systems [3], central to optimization and minimization of queries [5] and to data integration problems, such as view-based query answering [25, 27].

Conjunctive queries (CQs, for short) [5] are a restricted form of *first-order formulas* and correspond to the *select-project-join* fragment of relational algebra [12, 13], which is at the core of all major structured query languages, like SQL [1] and SPARQL [18]. Such types of queries have been extensively investigated by the database and artificial intelligence communities. In particular, in their breakthrough article [5], Chandra and Merlin solved the decidability of the containment problem for CQs, by proving that it is essentially the same problem as CQ evaluation, and it amounts to finding an *homomorphism* that syntactically maps one query into the other. This places the problem in NPTime, and it is hard for the same class as well.

Although widely used as a language abstraction for SQL analysis, the classic *set-theoretic semantics* of CQs does not precisely capture the actual interpretation of SQL queries on relational databases. Under the standard semantics, indeed, both the answer to a CQ and all relations in the underlying database are modeled as sets of tuples. In practice, however, database systems allow for multiple replicas of a tuple in their tables. The answers to a query might also contain several

occurrences of the same result. Motivated by this mismatch, Chaudhuri and Vardi [7] posed the question of deciding containment of CQs under *bag* or *multiset semantics*, where the answer and the database tables are collections of possibly replicated tuples, *i.e.*, *bags*. In their seminal work, they claimed that the relevant problem, referred to as *bag containment*, is Π_2^{p}-hard, but its exact complexity, even worst its decidability, is unknown to date.

Several attacks on this problem (including two PhD theses) have failed [24]. However, there have been successful attempts, some of them reported in the following, to prove un-/decidability or complexity results for variations of the problem, usually focusing on extensions or restrictions of the CQ language. On one side, Ioannidis and Ramakrishnan [20] remarkably proved that the bag-containment problem for *unions of* CQ*s* (UCQs, for short) is undecidable by applying a reduction from the *Diophantine inequality problem*, which is a variant of the well-known *Hilbert's 10th problem* [28] that, in turn, is about deciding whether a polynomial equation with integer coefficients has an all-integer solution. In more detail, they described a technique to derive, from two arbitrary polynomials $P_1(\overline{u})$ and $P_2(\overline{u})$, with natural coefficients, no constant terms, and same unknowns $\overline{u}$, two UCQs q_1 and q_2 such that q_1 is bag contained into q_2, in symbols $q_1 \sqsubseteq_{\mathsf{b}} q_2$, *iff* $P_1(\overline{\xi}) \leq P_2(\overline{\xi})$, for all vectors of natural numbers $\overline{\xi} \in \mathbb{N}^{|\overline{u}|}$. By reducing from another variant of the same problem, Jayram *et al.* [21] showed that bag containment for CQ*s with inequalities* is also undecidable. On the other side, Kopparty and Rossman [26] proved the decidability of bag containment for a class of *Boolean* CQ*s* that enjoys certain graph-theoretic properties when syntactically interpreted as graphs. Their results exploit information-theory techniques based on entropy measures, but do not provide explicit complexity bounds. Finally, Afrati *et al.* [2] reported decidability and complexity results for five major subclasses of CQs that have important practical applications.

Apart form the original question, the major open problem out of those that Afrati *et al.* pointed out in their article is the bag containment of a *projection-free* CQ, *i.e.*, a conjunctive query with no existential variables, into a CQ that might have projections. In the present work, we solve this problem by providing a Π_2^{p} decidability procedure. Different from all other approaches aiming at a decidability result and inspired by the converse techniques of those exploited in the proofs of undecidability, we describe an exponential reduction to the solution of a Diophantine inequality system of a certain structure. The latter is then shown to be decidable in PTime. In more detail, given a projection-free CQ q_1 and a generic CQ q_2, we first construct a family of monomial-polynomial pairs $\{(M_i(\overline{u}), P_i(\overline{u}))\}_{i=1}^{k}$, all with natural coefficients and no constant terms, such that $q_1 \sqsubseteq_{\mathsf{b}} q_2$ *iff* $M_i(\overline{\xi}) \leq P_i(\overline{\xi})$, for all $\overline{\xi} \in \mathbb{N}^{|\overline{u}|}$ and $1 \leq i \leq k$. Both the number and the size of the polynomials $P_i(\overline{u})$ might be exponential in the size of the containing query q_2, but are polynomial in those of the containee query q_1. Then, we linearly reduce the solution of each converse *monomial-polynomial inequality* $M_i(\overline{u}) > P_i(\overline{u})$ to the search for a solution of a suitable *linear inequality system*. The latter problem is well-known to be computable in PTime [29, 32]. Finally, we characterize the complexity of bag containment by relating it to the second level of the polynomial hierarchy, since it enjoys a witness whose correctness can be checked by a $\forall\exists$-alternating Turing machine. As lower bound, we provide an NPTime-hardness result.

We conclude by discussing possible generalizations of our approach to more expressive "containee" queries, pointing out difficulties for the general case. We hope this enables discussions for future research opportunities in the area.

2 PRELIMINARIES

A *bag* over a set I is a function $\mu\colon \mathrm{I} \to \mathbb{N}$ mapping every element $t \in \mathrm{I}$ to a non-negative integer value $\mu(t)$, called *multiplicity* of t *w.r.t.* μ in I, and it is *finite* if I is finite. We shall be writing a bag μ over a set I as the expression $\mathrm{I}^{\mu} = \left\{t^{\mu(t)} : t \in \mathrm{I}\right\}$. For two bags μ_1 and μ_2 over the sets I_1 and I_2, respectively, we say that μ_1 is a *subbag* of μ_2, in symbols $\mu_1 \subseteq \mu_2$, if $\mathrm{I}_1 \subseteq \mathrm{I}_2$ and $\mu_1(t) \leq \mu_2(t)$, for all $t \in \mathrm{I}_1$.

We use the standard mathematical logic notation of *relation*, *constant*, *variable*, and *term*, the latter denoting either a constant or a variable. An n-tuple $\overline{t}$ is an ordered list of n terms, with n its *arity* that can be also indicated by $|\overline{t}|$. Relations are also associated with a non-negative number called arity. A relation name R together with its arity n forms a *relation schema*. *Atoms* are syntactic expressions of the form $R(\overline{t})$, where R is a relation name having arity n and $\overline{t}$ is an n-tuple of terms. *Ground atoms*, *a.k.a. facts*, are atoms $R(\overline{c})$ where $\overline{c}$ is a tuple of constants only. Therefore, a relation (or relation instance) over relation schema R is essentially a set of facts $R(\overline{c})$. X and C denote the set of variables and constants, respectively. Sometimes it is convenient to interpret a variable (*resp.*, a tuple of variables) as constant (*resp.*, a tuple of constants). We call these *canonical constants* (disjoint from the set of constants in our language) and define a bisection between the canonical constant $\widehat{x}$ and the associated variable x (see [1]). Formally, $\mathrm{C} = \mathrm{C}_c \cup \mathrm{C}_l$, $\mathrm{C}_c \cap \mathrm{C}_l = \emptyset$, where C_c and C_l are the sets of canonical and language constants, respectively. Moreover, for a set of variables $\mathrm{V} \subseteq \mathrm{X}$, we have $\mathsf{can}(\mathrm{V}) \triangleq \{\widehat{x} : x \in \mathrm{V}\} \subseteq \mathrm{C}_c$. Note that $\mathrm{C}_c = \mathsf{can}(\mathrm{X})$.

A set of relation schemas is a *database schema* S. Accordingly, a *set database instance* (*set instance*, for short) I is a (possibly infinite) set of facts belonging in relation instances over relation schemas in S. The *active domain* $\mathsf{adom}(\mathrm{I})$ of a

set database instance I is the set of all constants that occur in I. A bag over a set instance is a *bag instance*.

A *substitution* σ is a mapping of variables to constants. We write $\sigma = \{x_1 \mapsto c_1; \ldots; x_n \mapsto c_n\}$ to indicate that $\sigma(x_i) = c_i$, for $1 \leq i \leq n$. For α a term, a tuple of terms, an atom, or a set of atoms, $\sigma(\alpha)$ is obtained by replacing each occurrence of a variable x in α, that also occurs in the domain of σ, with $\sigma(x)$; variables outside the domain of σ remain unchanged. Given $\overline{x}$ an n-tuple of variables $x_1, \ldots, x_n$ and $\overline{t}$ an n-tuple of constants $t_1, \ldots, t_n$, we say that $\overline{x}$ and $\overline{t}$ *unify* with each other or are *unifiable*, if we can define the substitution σ in such a way that $\sigma(\overline{x}) = \overline{t}$. A substitution σ is a *homomorphism* of a set of atoms I_1 into a set of atoms I_2, if the domain of σ is the set of all variables occurring in I_1 and $\sigma(\mathrm{I}_1) \subseteq \mathrm{I}_2$.

A *conjunctive query* (CQ, for short) $q(\overline{x})$ with $\overline{x}$ the tuple of *free variables* (also denoted by q when we do not focus on its free variables) is a first-order formula of the form $\exists \overline{y} \bigwedge_i R_i(\overline{x}, \overline{y})$, where $\bigwedge_i R_i(\overline{x}, \overline{y})$ is a conjunction of atoms with variables from vector $\overline{xy}$. When $\overline{y}$ is empty, $q(\overline{x})$ is called a *projection-free* CQ. We usually use the datalog notation

$$q(\overline{x}) \leftarrow R_1(\overline{x}, \overline{y}), \ldots, R_n(\overline{x}, \overline{y})$$

in place of $q(\overline{x}) = \exists \overline{y} \bigwedge_{i=1}^{n} R_i(\overline{x}, \overline{y})$. As a syntactic object, the query $\exists \overline{y} \bigwedge_i R_i(\overline{x}, \overline{y})$ might contain repeated atoms, that is, there might be indexes i, j with $i \neq j$ such that $R_i(\overline{x}, \overline{y}) = R_j(\overline{x}, \overline{y})$. Traditionally, in set semantics, this repetition does not make any difference since it does not affect the result of a query. Under the bag perspective, however, we want to model these repetitions. The body of a CQ $q(\overline{x}) = \exists \overline{y} \bigwedge_i R_i(\overline{x}, \overline{y})$ is the set $\mathrm{body}(q(\overline{x}))$ that contains all distinct atoms in $\bigwedge_i R_i(\overline{x}, \overline{y})$. We can then define a bag $\mu_{q(\overline{x})}$ over $\mathrm{body}(q(\overline{x}))$, which given an atom in the body returns the number of occurrences of this atom in the expression $\bigwedge_i R_i(\overline{x}, \overline{y})$. Bag $\mu_{q(\overline{x})}$ is called the *body multiplicity* of $q(\overline{x})$. Any CQ can now be seen as the pair $\langle \overline{x}, \mu_{q(\overline{x})} \rangle$, which is called the *bag representation* of the corresponding CQ. Consider, for example, the query

$$q(x_1, x_2) \leftarrow R(x_1, y_1), R(x_1, y_1), R(x_1, y_2), P(y_2, y_3), P(y_2, y_3), P(x_2, y_4).$$

The bag representation of this CQ is $\langle x_1 x_2, \mu_q \rangle$, where μ_q is defined over the body

$$\mathrm{body}(q) = \{R(x_1, y_1), R(x_1, y_2), P(y_2, y_3), P(x_2, y_4)\},$$

with $\mu_q(R(x_1, y_1)) = \mu_q(R(y_2, y_3)) = 2$ and $\mu_q(R(x_1, y_2)) = \mu_q(R(x_2, y_4)) = 1$. To ease presentation, Table 1 reports a summary of our notations and examples of their usage.

We often write the bag representation of a conjunctive query by giving the datalog notation of the body with superscripts on the atoms in order to annotate them with their multiplicity. For instance, the bag representation of the previous example is

$$q(x_1, x_2) \leftarrow R^2(x_1, y_1), R^1(x_1, y_2), P^2(y_2, y_3), P^1(x_2, y_4).$$

Sometimes, we also omit the superscripts 1 as in

$$q(x_1, x_2) \leftarrow R^2(x_1, y_1), R(x_1, y_2), P^2(y_2, y_3), P(x_2, y_4).$$

For the sake of compactness, we might also write

$$\mathrm{body}(q) = \{R^2(x_1, y_1), R(x_1, y_2), P^2(y_2, y_3), P(x_2, y_4)\}.$$

The *active domain* $\mathrm{adom}(q)$ of a query q is the set of all constants that occur in it. The set of all variables in q is denoted $\mathrm{var}(q)$. The *canonical database instance* of q, denoted by I^q, is the set of facts obtained by replacing all variables x in $\mathrm{body}(q)$ with the associated canonical constant $\widehat{x}$ (see [1]).

For a given query q and substitution σ, the symbol $\sigma(q)$ denotes the query obtained by applying the substitution σ to each single atom in the body $\mathrm{body}(q)$. Informally, any repetition of atoms in q is carried over in $\sigma(q)$ and, in addition, the latter could result with more repeated atoms. Formally, $\sigma(q(\overline{x}))$ is a CQ with free variables $\sigma(\overline{x})$, body $\mathrm{body}(\sigma(q(\overline{x}))) = \sigma(\mathrm{body}(q(\overline{x})))$, and body multiplicity $\mu_{\sigma(q(\overline{x}))}$ defined as follows:

$$\mu_{\sigma(q(\overline{x}))}(\alpha) = \sum_{\beta \in \mathrm{body}(q(\overline{x}))}^{\sigma(\beta)=\alpha} \mu_{q(\overline{x})}(\beta). \tag{1}$$

For instance, consider the substitution $\sigma = \{y_1, y_2, y_3, y_4 \mapsto x_2\}$ and the query q of the above example. The bag representation of $\sigma(q)$ is

$$\sigma(q)(x_1, x_2) \leftarrow R^3(x_1, x_2), P^3(x_2, x_2).$$

When a tuple of free variables $\overline{x}$ is unifiable with a tuple of constants $\overline{t}$ via the substitution σ, the symbol $q(\overline{t})$ denotes the result of $\sigma(q(\overline{x}))$, *i.e.*, the Boolean query obtained by replacing $\overline{x}$ with $\overline{t}$ in q.

The *answer under set semantics* to a CQ $q(\overline{x})$ over an instance I, in symbols $q(\overline{x})^{\mathrm{I}}$ or q^{I} when the tuple of free variables is not important, is the set of all tuples $\overline{c} \in \mathrm{adom}(\mathrm{I})^{|\overline{x}|}$ (tuples of constants from I) unifiable with $\overline{x}$, such that query $q(\overline{c})$ holds in I, *i.e.*, $\mathrm{body}(q(\overline{c})) \subseteq I$. For every answer tuple $\overline{c}$ of $q(\overline{x})$ over I, there is a homomorphism h of $\mathrm{body}(q(\overline{x}))$ into I such that $\mathsf{h}(\overline{x}) = \overline{c}$. We often say that such function h is a homomorphism of q into I and use $\mathrm{Hom}(q(\overline{x}), \mathrm{I})$ to denote the associated set of all these functions.

Let μ be a bag over a set instance I. The *answer under bag semantics* to a CQ $q(\overline{x}) = \exists \overline{y} \bigwedge_i R_i(\overline{x}, \overline{y})$ over μ, is the bag over $\mathrm{adom}(\mathrm{I})^{|\overline{x}|}$, denoted by $q^{\mu}(\overline{x})$, such that, for all tuples $\overline{c} \in \mathrm{adom}(\mathrm{I})^{|\overline{x}|}$ unifiable with $\overline{x}$, it holds that

$$q^{\mu}(\overline{c}) = \sum_{\mathsf{h} \in \mathrm{Hom}(q(\overline{x}), \mathrm{I})}^{\mathsf{h}(\overline{x})=\overline{c}} \prod_{\alpha \in \mathrm{body}(\mathsf{h}(q(\overline{x})))} \mu(\alpha)^{\mu_{\mathsf{h}(q(\overline{x}))}(\alpha)} \tag{2}$$

Note that, for all elements $\overline{c} \notin q(\overline{x})^{\mathrm{I}}$, we have that $q^{\mu}(\overline{c}) = 0$, since the sum in Equation 2 ranges over an empty set of homomorphisms, *i.e.*, there are no homomorphisms h as

Table 1: Common notations.

Notation	Meaning	Example Usage
$\overline{t_i}$	a tuple of terms	$h(\overline{t_1}) = \overline{t_2}$
q or $q(\overline{x})$	a query (the semantic object)	$q_1 \sqsubseteq_s q_2$
$q(\overline{t})$	query obtained by replacing terms $\overline{x}$ in $q(\overline{x})$ with terms $\overline{t}$	$q(\overline{c})$ is Boolean if $\overline{c}$ is a tuple of constants, and in addition, it is ground if q is projection free
μ	a bag, i.e., a function $\mu\colon \mathrm{I} \to \mathbb{N}$ over a set I	a bag μ over a set instance I is a bag instance; for a tuple $\overline{t} \in \mathrm{I}$, $\mu(\overline{t}) > 0$
I^μ	convenience notation for μ, listing elements in I and multiplicities, i.e., $\mathrm{I}=\{t^{\mu(t)} : t \in \mathrm{I}\}$	if μ_1 and μ_2 both over I, and $\mu_1 \subseteq \mu_2$, then for all $\langle t^i, t^j\rangle \in I^{\mu_1} \times I^{\mu_2}$, $i \leq j$
$\exists \overline{y} \bigwedge_{i=1}^n R_i(\overline{x}, \overline{y})$	conj. queries in FOL	when $\overline{y} = \emptyset$, $\bigwedge_{i=1}^n R_i(\overline{x}, \overline{y})$ is projection free
$q(\overline{x}) \leftarrow R_1(\overline{x}, \overline{y}), \ldots, R_n(\overline{x}, \overline{y})$	conj. queries in datalog	$body(q) = \{R_1(\overline{x}, \overline{y}), \ldots, R_n(\overline{x}, \overline{y})\}$
q^{I}	the set answer of a query q over a set instance I	$q_1 \sqsubseteq_s q_2$ iff for all set instances J, $q_1^J \subseteq q_2^J$
I^q or $I^{q(\overline{x})}$	the canonical instance of q	$\overline{t}$ is an answer tuple of $q(\overline{x})$ over $\mathrm{I}^{q(\overline{t})}$
$\mu_{q(\overline{x})}$	body multiplicity of query $q(\overline{x})$, i.e., a bag over $body(q)$	if atom α appears twice in $q(\overline{x}) = \exists \overline{y} \bigwedge_{i=1}^n R_i(\overline{x}, \overline{y})$, then $\mu_{q(\overline{x})}(\alpha) = 2$
$\langle \overline{x}, \mu_{q(\overline{x})}\rangle$	bag representation of a query $q(\overline{x})$	$\langle \emptyset,\ \{R(\overline{x}) \rightarrow 2\}\rangle$ is the bag representation of Boolean query $\exists \overline{x} R(\overline{x}) R(\overline{x})$
q^μ or $q^\mu(\overline{x})$	a bag answer of $q(\overline{x})$ over μ	$(q^I)^{q^\mu} = \{\overline{t}^{q^\mu(\overline{t})} : \overline{t} \in q^{\mathrm{I}}\} \cup \{\overline{t}^0 : \overline{t} \notin q^{\mathrm{I}}, \overline{t} \in adom(I)^{\lvert\overline{x}\rvert}\}$

required. Thus, by restricting q^μ over $q(\overline{x})^{\mathrm{I}}$, we sometimes abuse the notation and represent the answer q^μ as

$$\left\{\overline{c}^{q^\mu(\overline{c})} : \overline{c} \in q(\overline{x})^{\mathrm{I}}\right\}.$$

To practice with the introduced notions consider the following example on the instance

$$\mathrm{I} = \{R(c_1, c_2), R(c_1, c_3), P(c_2, c_4), P(c_5, c_4)\}$$

and bag μ over it, such that

$$\mathrm{I}^\mu = \{R^2(c_1, c_2), R^1(c_1, c_3), P^1(c_2, c_4), P^3(c_5, c_4)\}.$$

Also, consider again the query from the previous example

$$q(x_1, x_2) \leftarrow R^2(x_1, y_1), R(x_1, y_2), P^2(y_2, y_3), P(x_2, y_4).$$

The answer under bag semantics to $q(x_1, x_2)$ over μ is

$$q^\mu = \{c_1 {c_2}^{10}, c_1 {c_5}^{30}\},$$

i.e., $q^\mu(c_1, c_2) = 10$ and $q^\mu(c_1, c_5) = 30$. To see why this is true, first notice that any homomorphism of $q(x_1, x_2)$ into I must map the two variables y_3 and y_4 to the constant c_4 and the variable y_2 to a constant which belongs to both a fact in R and one in P, and c_2 is the only value with this property. Hence, any such a homomorphism must be an extension of

$$\{x_1 \mapsto c_1; y_2 \mapsto c_2; y_3, y_4 \mapsto c_4\}.$$

For c_1c_2, *i.e.*, when $x_2 \mapsto c_2$, we have two homomorphisms that give us this answer tuple:

$$\mathsf{h}_1 = \{x_1 \mapsto c_1; x_2, y_1, y_2 \mapsto c_2; y_3, y_4 \mapsto c_4\};$$
$$\mathsf{h}_2 = \{x_1 \mapsto c_1; x_2, y_2 \mapsto c_2; y_1 \mapsto c_3; y_3, y_4 \mapsto c_4\}.$$

Hence, according to Equation 2, we have

$$q^\mu(c_1, c_2) = \prod_{\alpha \in \mathrm{body}(\mathsf{h}_1(q(x_1, x_2)))} \mu(\alpha)^{\mu_{\mathsf{h}_1(q(x_1, x_2))}(\alpha)} + \prod_{\alpha \in \mathrm{body}(\mathsf{h}_2(q(x_1, x_2)))} \mu(\alpha)^{\mu_{\mathsf{h}_2(q(x_1, x_2))}(\alpha)}.$$

The bag representation of $\mathsf{h}_1(q(x_1, x_2))$ is

$$q(c_1, c_2) \leftarrow R^3(c_1, c_2), P^3(c_2, c_4),$$

while and the bag representation of $\mathsf{h}_2(q(x_1, x_2))$ is

$$q(c_1, c_2) \leftarrow R(c_1, c_2), R^2(c_1, c_3), P^3(c_2, c_4).$$

Hence, the above equation simplifies in

$$q^\mu(c_1, c_2) = 2^3 \times 1^3 + 2 \times 1^2 \times 1^3 = 10.$$

Similarly, for answer tuple c_1c_5, we have two homomorphisms from $q(x_1, x_2)$ to I:

$$\mathsf{h}_3 = \{x_1 \mapsto c_1; x_2 \mapsto c_5; y_1, y_2 \mapsto c_2; y_3, y_4 \mapsto c_4\};$$
$$\mathsf{h}_4 = \{x_1 \mapsto c_1; x_2 \mapsto c_5; y_1 \mapsto c_3; y_2 \mapsto c_2; y_3, y_4 \mapsto c_4\}.$$

The multiplicity of c_1c_5 is computed according to Equation 2 by means of the formula

$$q^{\mu}(c_1, c_5) = \prod_{\alpha \in \mathsf{body}(\mathsf{h}_3(q(x_1,x_2)))} \mu(\alpha)^{\mu_{\mathsf{h}_3(q(x_1,x_2))}(\alpha)} + \prod_{\alpha \in \mathsf{body}(\mathsf{h}_4(q(x_1,x_2)))} \mu(\alpha)^{\mu_{\mathsf{h}_4(q(x_1,x_2))}(\alpha)}.$$

For the bag representation of $\mathsf{h}_3(q(x_1, x_2))$, we have

$$q(c_1, c_5) \leftarrow R^3(c_1, c_2), P^2(c_2, c_4), P(c_5, c_4),$$

while $\mathsf{h}_4(q(x_1, x_2))$ has as bag representation

$$q(c_1, c_5) \leftarrow R(c_1, c_2), R^2(c_1, c_3), P^2(c_2, c_4), P(c_5, c_4).$$

Thus, the previous equation becomes

$$q^{\mu}(c_1, c_5) = 2^3 \times 1^2 \times 3 + 2 \times 1^2 \times 1^2 \times 3 = 30.$$

Given two CQs q_1 and q_2 we say that q_1 is set contained in q_2, in symbols $q_1 \sqsubseteq_s q_2$, if, for all instances I, the set of answer tuples under set semantics of q_1 on I is a subset of the answer tuples under set semantics of q_2 on I, that is $q_1^{\mathrm{I}} \subseteq q_2^{\mathrm{I}}$. Similarly, we say that q_1 is bag contained in q_2, in symbols $q_1 \sqsubseteq_b q_2$, if for all set instances I and bags μ over I, it holds that $q_1^{\mu} \subseteq q_2^{\mu}$. When discussing about a containment problem $q_1 \sqsubseteq_b q_2$, we shall refer to q_1 as the "*containee*" query and to q_2 as the "*containing*" one, even if the containment does not actually hold between two particular queries.

Chandra and Merlin [5] have proved that one can decide set query containment in NPTime, by using the notion of *containment mappings*. Formally, a containment mapping from a CQ $q_2(\overline{x_2})$ to a CQ $q_1(\overline{x}_1)$ is a homomorphism $\mathsf{h} \in \mathrm{Hom}(\mathsf{body}(q_2(\overline{x}_2)), \mathsf{body}q_1(\overline{x}_1))$ from $q_2(\overline{x}_2)$ to $q_1(\overline{x}_1)$ such that $\mathsf{h}(\overline{x}_2) = \overline{x}_1$. Then, we have that $q_1 \sqsubseteq_s q_2$ *iff* there exists a containment mapping from $q_2(\overline{x_2})$ to $q_1(\overline{x}_1)$. Later in this work we make use of the set of all containment mappings from $q_2(\overline{x_2})$ to $q_1(\overline{x}_1)$ denoted by $\mathrm{CM}(q_2(\overline{x}_2), q_1(\overline{x}_1))$. Moreover, for a given tuple of terms $\overline{t}$, we make an abuse of notation by using $\mathrm{CM}(q_2(\overline{x}_2), q_1(\overline{t}))$ to represent the set of all compositions of a containment mapping $\mathsf{h} \in \mathrm{CM}(q_2(\overline{x}_2), q_1(\overline{x}_1))$ with a substitution σ such that $\sigma(\mathsf{h}(\overline{x}_2)) = \overline{t}$.

Observe that bag containment implies set containment, *i.e.*, for all pairs of queries $q_1(\overline{x_1})$ and $q_2(\overline{x}_2)$, if $q_1 \sqsubseteq_b q_2$ then $q_1 \sqsubseteq_s q_2$. This can be easily seen by unfolding the definitions. Indeed, if all tuples that appear in the answer of $q_1(\overline{x_1})$ with some multiplicity also appear in the answer of $q_2(\overline{x}_2)$ with at least the same multiplicity, then set containment holds by simply ignoring the multiplicities.

As an example of set and bag containment, consider the following three queries:

- $q_1(x_1, x_2) \leftarrow R^2(x_1, x_2), P^3(x_2, x_2)$;
- $q_2(x_1, x_2) \leftarrow R^3(x_1, x_2), P^3(x_2, x_2)$;
- $q_3(x_1, x_2) \leftarrow R^2(x_1, y_1), R(x_1, y_2), P^2(y_2, y_3), P(x_2, y_4)$.

It is not hard to see that:

(1) $q_1 \sqsubseteq_b q_2$ and $q_2 \sqsubseteq_s q_1$, but $q_2 \not\sqsubseteq_b q_1$;
(2) $q_1 \sqsubseteq_b q_3$ and $q_2 \sqsubseteq_b q_3$ (so, $q_1 \sqsubseteq_s q_3$ and $q_2 \sqsubseteq_s q_3$);
(3) $q_3 \not\sqsubseteq_s q_1$ and $q_3 \not\sqsubseteq_s q_2$ (so, $q_3 \not\sqsubseteq_b q_1$ and $q_3 \not\sqsubseteq_b q_2$).

Let us see why these statements hold.

(1) The identity substitution is the only containment mapping from $q_1(x_1, x_2)$ to $q_2(x_1, x_2)$ and *vice versa*. Moreover, every multiplicity of an atom, in an arbitrary instance, used as an image of $R(x_1, y_1)$, is raised to 2 in q_1 and 3 in q_2. So the bag containment of the first query into the second one trivially follows. To see why $q_2 \not\sqsubseteq_b q_1$, it is enough to observe that, on the bag instance $\mathrm{I}^{\mu} = \{R^2(c_1, c_2), P^1(c_2, c_2)\}$, we have $q_1^{\mu}(c_1, c_2) = 4$, but $q_2^{\mu}(c_1, c_2) = 8$.
(2) The substitution $\sigma = \{y_1, y_2, y_3, y_4 \mapsto x_2\}$ is the unique containment mapping of $q_3(x_1, x_2)$ to both $q_1(x_1, x_2)$ and $q_2(x_1, x_2)$. Since $\sigma(q_3(x_1, x_2)) = q_2(x_1, x_2)$, the statement follows from above.
(3) There are no containment mappings from $q_1(x_1, x_2)$ and $q_2(x_1, x_2)$ to $q_3(x_1, x_2)$.

3 A BAG-CONTAINMENT PROBLEM

In this section we start studying the bag-containment problem of a projection-free CQ into a generic CQ. In particular, we show that in order to verify that containment holds, it suffices to evaluate the two queries on a special class of infinitely many *canonical bags*. In other words, we use a particular class of canonical set instances and check the containment on all possible bags over all such instances. Note that this problem is trivial in the so-called bag-set semantics [2], since it is equivalent to set-containment, but requires non-trivial treatment in our case, as we show in the rest of the article.

The use of more than one canonical set databases, and in fact, exponentially many endomorphic versions of the canonical database of the containee query, has been already employed in query containment before, as in [23] for testing containment of CQs with arithmetic comparisons and in [34] for testing containment of CQs with negation. The set of instances that has been used in these approaches is obtained by taking the canonical instances of all queries that are "specializations" of the original containee query, *i.e.*, where some of its free variables have been equated to each other.

For our purposes, we exploit a variation of this idea and use all canonical instances where some of the free variables have been equated to each other or replaced with constants of the original query. In order to create such a set of instances, we shall ground the containee query $q(\overline{x})$, by substituting the free variables $\overline{x}$ with an appropriate *probe tuple* $\overline{t}$, whose terms are chosen from the canonical constants in $\mathsf{can}(\overline{x})$, as well as the preexisting constants in $\mathsf{adom}(q(\overline{x}))$. Obviously, $\overline{x}$ and $\overline{t}$ need to be unifiable, so that the query $q(\overline{t})$ derivable from the grounding is well-defined.

Definition 3.1. *Let $q(\overline{x})$ be a* CQ *defined over an n-tuple of free variables $\overline{x} \in \mathrm{X}^n$. An n-tuple of constants $\overline{t} \in \mathrm{C}^n$ is a* probe tuple *for $q(\overline{x})$ if (1) $\overline{t} \in \mathsf{adom}(\mathrm{I}^{q(\overline{x})})^n$ and (2) $\overline{t}$ is unifiable with $\overline{x}$.*

In the following, we use $\mathsf{prbtup}(q(\overline{x}))$ to denote the set of all probe tuples for a CQ $q(\overline{x})$.

As an example, consider the projection-free CQ

$$q(x_1, x_2) \leftarrow R(x_1, x_2), R(\mathsf{c}_1, x_2), R(x_1, \mathsf{c}_2).$$

It is not hard to see that there are sixteen probe tuples for $q(x_1, x_2)$, *i.e.*, all possible pairs over the elements $\{\widehat{x}_1, \widehat{x}_2, \mathsf{c}_1, \mathsf{c}_2\}$:

$$\left\{ \begin{array}{c} \widehat{x}_1\widehat{x}_1, \widehat{x}_1\widehat{x}_2, \widehat{x}_2\widehat{x}_1, \widehat{x}_2\widehat{x}_2, \\ \widehat{x}_1\mathsf{c}_1, \widehat{x}_1\mathsf{c}_2, \widehat{x}_2\mathsf{c}_1, \widehat{x}_2\mathsf{c}_2, \mathsf{c}_1\widehat{x}_1, \mathsf{c}_1\widehat{x}_2, \mathsf{c}_2\widehat{x}_1, \mathsf{c}_2\widehat{x}_2, \\ \mathsf{c}_1\mathsf{c}_1, \mathsf{c}_1\mathsf{c}_2, \mathsf{c}_2\mathsf{c}_1, \mathsf{c}_2\mathsf{c}_2 \end{array} \right\}.$$

It is possible to prove that for our technique we do not need all probe tuples, but only those that are not isomorphic under a bijective function that preserves the language constants, but swaps the canonical constants. For the given example, we would only need the following ten tuples:

$$\left\{ \begin{array}{c} \widehat{x}_1\widehat{x}_1, \widehat{x}_1\widehat{x}_2, \\ \widehat{x}_1\mathsf{c}_1, \widehat{x}_1\mathsf{c}_2, \mathsf{c}_1\widehat{x}_1, \mathsf{c}_2\widehat{x}_1, \\ \mathsf{c}_1\mathsf{c}_1, \mathsf{c}_1\mathsf{c}_2, \mathsf{c}_2\mathsf{c}_1, \mathsf{c}_2\mathsf{c}_2 \end{array} \right\}.$$

However, for the sake of simplicity in the presentation of the results we do not make use of the restricted set.

At this point, to characterize the bag-containment problem of interest, we make use of the specialized queries $q_1(\overline{t})$ and their canonical instances $\mathrm{I}^{q_1(\overline{t})}$, for each probe tuple $\overline{t}$ of the given projection-free containee query $q_1(\overline{x}_1)$. Next theorem states, indeed, that $q_1(\overline{x}_1)$ is bag contained into a generic CQ $q_2(\overline{x}_2)$ *iff*, for all possible bags μ over all such specialized canonical instances $\mathrm{I}^{q_1(\overline{t})}$, the multiplicity of $\overline{t}$ in the answer of $q_1(\overline{x}_1)$ over μ is less than or equal to that of $q_2(\overline{x}_2)$ over μ as well. In other words, if containment breaks for an arbitrary answer tuple over an arbitrary bag, there must be a probe tuple $\overline{t}$ and some bag μ over $\mathrm{I}^{q_1(\overline{t})}$ where containment breaks over this bag μ and for that particular tuple $\overline{t}$.

Theorem 3.1. *Given a projection-free* CQ $q_1(\overline{x}_1)$ *and a* CQ $q_2(\overline{x}_2)$*, it holds that $q_1(\overline{x}_1) \sqsubseteq_b q_2(\overline{x}_2)$* iff*, for all probe tuples $\overline{t} \in \mathsf{prbtup}(q_1(\overline{x}_1))$, (1) $\overline{t}$ is unifiable with $\overline{x}_2$ and (2) $q_1^{\mu}(\overline{t}) \leq q_2^{\mu}(\overline{t})$, for all bags μ over the canonical instance $\mathrm{I}^{q_1(\overline{t})}$.*

Proof. **(Only if).** Assume $q_1(\overline{x}_1) \sqsubseteq_b q_2(\overline{x}_2)$. Then, $q_1^{\mu}(\overline{x}_1) \subseteq q_2^{\mu}(\overline{x}_2)$, for all instances I and bags μ over I. This means that, for all tuples $\overline{c} \in q_1^{\mathrm{I}}$, it holds that (i) $\overline{c}$ is unifiable with $\overline{x}_2$ and (ii) $q_1^{\mu}(\overline{c}) \leq q_2^{\mu}(\overline{c})$. Now, consider the canonical instance $\mathrm{I}^{q_1(\overline{t})}$ for $q_1(\overline{x}_1)$, a bag μ over it, and a probe tuple $\overline{t} \in \mathsf{prbtup}(q_1(\overline{x}_1)) \subseteq \mathsf{adom}(\mathrm{I}^{q_1(\overline{t})})^{|\overline{x}_1|}$. It is immediate to see that $q_1^{\mu}(\overline{t}) \leq q_2^{\mu}(\overline{t})$ holds. Hence, the thesis follows.

(If). Assume $q_1(\overline{x}_1) \not\sqsubseteq_b q_2(\overline{x}_2)$. This means that there exists an instance I and a bag μ over I such that $q_1^{\mu}(\overline{x}_1) \not\subseteq q_2^{\mu}(\overline{x}_2)$. Therefore, there has to exist a tuple $\overline{c} \in q_1^{\mathrm{I}}$ satisfying one of the following properties: (i) $\overline{c}$ is not unifiable with $\overline{x}_2$ or (ii) $\overline{c}$ is unifiable with $\overline{x}_2$, but $q_1^{\mu}(\overline{c}) > q_2^{\mu}(\overline{c})$. In the first case, the thesis is immediately, so, focus on the second one.

Since $q_1^{\mu}(\overline{c}) > q_2^{\mu}(\overline{c})$, we have that $q_1^{\mu}(\overline{c}) > 0$, which implies $\overline{c} \in q_1^{\mathrm{I}}(\overline{x}_1)$, *i.e.*, there exists an homomorphism h_1 from $q_1(\overline{x}_1)$ to I mapping $\overline{x}_1$ to $\overline{c}$, *i.e.*, $\mathsf{h}_1(\overline{x}_1) = \overline{c}$. Being projection-free, $q_1(\overline{x}_1)$ does not have existential variables, thus, h_1 is the only such homomorphism. Moreover, the image of the query under h_1, *i.e.*, $\mathsf{h}_1(q_1(\overline{x}_1))$, is exactly $q_1(\overline{c})$ and its body is equal to $\mathrm{I}^{q_1(\overline{c})} \subseteq \mathrm{I}$, since its atoms do not contain variables. Now, let μ' be the restriction of the bag μ to $\mathrm{I}^{q_1(\overline{c})}$. Obviously, it holds that $q_1^{\mu}(\overline{c}) = q_1^{\mu'}(\overline{c})$. In addition, $q_2^{\mu}(\overline{c}) \geq q_2^{\mu'}(\overline{c})$. The latter inequality is due to the following facts: (i) all the homomorphisms h_2 from $q_2(\overline{x}_2)$ to $\mathrm{I}^{q_1(\overline{c})}$, with $\mathsf{h}_2(\overline{x}_1) = \overline{c}$, are homomorphisms from $q_2(\overline{x}_2)$ to I as well, since $\mathrm{I}^{q_1(\overline{c})} \subseteq \mathrm{I}$; (ii) the contribution, through Equation 2, of each such homomorphism h_2 to the multiplicity $q_2^{\mu}(\overline{c})$ of $\overline{c}$ over the (bag μ over) I is exactly the same as the one over (the bag μ' over) $\mathrm{I}^{q_1(\overline{c})}$; observe that there might even exist homomorphisms from $q_2(\overline{c})$ to $\mathrm{I} \setminus \mathrm{I}^{q_1(\overline{c})}$. Thus, $q_1^{\mu'}(\overline{c}) > q_2^{\mu'}(\overline{c})$.

At this point, consider a function $\mathsf{f} : \mathrm{C} \rightarrow \mathsf{adom}(\mathrm{I}^{q_1(\overline{x}_1)})$ mapping constants to the active domain of the canonical database of the containee query $q_1(\overline{x}_1)$ such that (i) it is the identity on the active domain of $q_1(\overline{x}_1)$, *i.e.*, $\mathsf{f}(c) = c$, for all $c \in \mathsf{adom}(q_1(\overline{x}_1))$, and (ii) its restriction $\mathsf{f}\restriction\mathsf{adom}(q_1(\overline{c}))$ to the active domain of the ground query $q_1(\overline{c})$ is injective. Intuitively, f injectively replaces all constant in $\overline{c}$ that do not occur in the body of $q_1(\overline{x}_1)$ with some of the canonical constants of the free variables $\overline{x}_1$, leaving untouched the remaining ones. It is easy to see that (i) $\overline{t} \triangleq \mathsf{f}(\overline{c}) \in \mathsf{prbtup}(q_1(\overline{x}_1))$ and (ii) f is an isomorphism between the canonical instances $\mathrm{I}^{q_1(\overline{c})}$ and $\mathrm{I}^{q_1(\overline{t})}$. Now, let us define the bag $\mu^{\star}$ over $\mathrm{I}^{q_1(\overline{t})}$ as follows: $\mu^{\star}(\mathsf{f}(\alpha)) \triangleq \mu'(\alpha)$, for all atoms $\alpha \in \mathrm{I}^{q_1(\overline{c})}$. It is obvious that $q_1^{\mu^{\star}}(\overline{t}) = q_1^{\mu'}(\overline{c})$ and $q_2^{\mu^{\star}}(\overline{t}) = q_2^{\mu'}(\overline{c})$. Consequently, $q_1^{\mu^{\star}}(\overline{t}) > q_2^{\mu^{\star}}(\overline{t})$. Hence, the thesis follows. □

Note that, given a CQ bag-containment problem, in particular a projection-free CQ $q_1(\overline{x}_1)$, as in Theorem 3.1, the set of tuples in $\mathsf{prbtup}(q_1(\overline{x}_1))$ is finite, and so is the set of our specialized canonical instances $\mathrm{I}^{q_1(\overline{t})}$. To decide containment, however, we need to reason over all possible, infinitely many, bags μ over these instances. It is therefore natural to assume the exact values of μ as unknowns and try to verify whether we can prove the required properties for such all bags together.

From now on, we assume a countably infinite set U of symbols called *unknowns*, disjoint from the sets of variables and

constants, and associate each possible atom with its "canonical" unknown in order to represent its unknown multiplicity. In more detail, we can assume a 1-to-1 correspondence between U and the countably infinite set of all possible atoms that can be constructed from relation names and having as arguments constants or variables. Through this correspondence, we associate each distinct atom α with its *canonical unknown* $u^\alpha \in \mathrm{U}$.

In the following, we shall use these unknowns in polynomial functions. In particular, we lift the above discussed association of atoms with unknowns to that of conjunctive queries with polynomial functions, so that the unknowns in the polynomials stand for the unknown values of a bag μ and the polynomial themselves stand for the multiplicity function of the answer tuples over μ.

First, we are going to associate a projection-free CQ $q(\overline{x})$ and a probe tuple $\overline{t} \in \mathrm{prbtup}(q(\overline{x}))$ with a monomial function $\mathsf{M}_{q(\overline{t})}(\overline{u})$. To do this, observe that there is only one homomorphism h, in fact an isomorphism, that maps $q(\overline{x})$ into the canonical instance $\mathrm{I}^{q(\overline{t})}$ such that $\mathsf{h}(\overline{x}) = \overline{t}$. This is precisely because $q(\overline{x})$ is projection-free. Thus, per Equation 2, the multiplicity of the tuple $\overline{t}$ in the answer of $q(\overline{x})$ over an arbitrary bag μ over the instance $\mathrm{I}^{q(\overline{t})}$ is:

$$q^\mu(\overline{t}) = \prod_{\alpha \in \mathrm{body}(\mathsf{h}(q(\overline{x})))} \mu(\alpha)^{\mu_{\mathsf{h}(q(\overline{x}))}(\alpha)} = \prod_{\alpha \in \mathrm{body}(q(\overline{t}))} \mu(\alpha)^{\mu_{q(\overline{t})}(\alpha)}.$$

Intuitively, if we interpret the values $\mu(\alpha)$ as unknowns, *i.e.*, we substitute $\mu(\alpha)$ with the canonical unknown u^α, we get the associated monomial for $q(\overline{x})$ and $\overline{t}$.

The *vector of unknowns* for a CQ $q(\overline{x})$ with $\mathrm{body}(q(\overline{x})) = \{\alpha_1, \alpha_2, \ldots, \alpha_m\}$, denoted by $\overline{u}^{q(\overline{x})}$, is the vector of canonical unknowns $\overline{u}^{q(\overline{x})} = u^{\alpha_1} u^{\alpha_2} \cdots u^{\alpha_m}$. For convenience, in our definitions, we want to be able to deal with vectors of unknowns which are supersequences of the vector of unknowns for a given specific query. We introduce the notion of compatibility of a vector of unknowns $\overline{u} \in \mathrm{U}^k$, for some $k \in \mathbb{N}$, with a query $q(\overline{x})$: we say that $\overline{u}$ is *compatible* with $q(\overline{x})$ if, for all $\alpha \in \mathrm{body}(q(\overline{x}))$, it holds that u^α occurs in $\overline{u}$.

Definition 3.2. *For all projection-free* CQs $q(\overline{x})$ *for which* $\mathrm{body}(q(\overline{x})) = \{\alpha_1, \ldots, \alpha_m\}$, *probe tuples* $\overline{t} \in \mathrm{prbtup}(q(\overline{x}))$ *unifiable with* $\overline{x}$, *and vectors of unknowns* $\overline{u} \in \mathrm{U}^k$ *compatible with* $q(\overline{x})$, *for some* $k \in \mathbb{N}$, *there is the associated* monomial

$$\mathsf{M}_{q(\overline{t})}(\overline{u}) \triangleq u_1^{e_1} \cdots u_k^{e_k},$$

where $e_i = \mu_{q(\overline{t})}(\alpha_i)$ *if* $u_i = u^{\alpha_i}$, *and* $e_i = 0$ *otherwise.*

As intuitively observed above, the monomial $\mathsf{M}_{q(\overline{t})}(\overline{u})$ allows us to compute the multiplicity of the answer $\overline{t}$ of the query $q(\overline{x})$ over any bag μ. Indeed, we just need to bind all unknowns u^{α_i} in $\overline{u}$ to the values given by μ for the corresponding atom $\alpha_i \in \mathrm{body}(q(\overline{x}))$, since the unknowns u_i that do not correspond to atoms in the body of query are cancelled out, as they are raised to 0.

To exemplify the above definition, consider the following bag variation of the projection-free CQ described in the example of the previous page:

$$q_1(x_1, x_2) \leftarrow R^2(x_1, x_2), R(\mathsf{c}_1, x_2), R^3(x_1, \mathsf{c}_2).$$

Moreover, consider probe tuple $\widehat{x}_1\widehat{x}_2 \in \mathrm{prbtup}(q_1(x_1, x_2))$. Then, it is easy to verify that

$$\mathsf{M}_{q_1(\widehat{x}_1,\widehat{x}_2)}(\overline{u}) = u_1^2 \cdot u_2 \cdot u_3^3, \quad \text{where}$$
$$u_1 = u^{R(\widehat{x}_1,\widehat{x}_2)}, u_2 = u^{R(\mathsf{c}_1,\widehat{x}_2)}, \text{ and } u_3 = u^{R(\widehat{x}_1,\mathsf{c}_2)},$$
$$\text{with} \quad q_1(\widehat{x}_1, \widehat{x}_2) \leftarrow R^2(\widehat{x}_1, \widehat{x}_2), R(\mathsf{c}_1, \widehat{x}_2), R^3(\widehat{x}_1, \mathsf{c}_2).$$

At this point, once a projection-free CQ $q_1(\overline{x}_1)$, a probe tuple $\overline{t}$, and vector of unknowns $\overline{u}$ are fixed, as in Definition 3.2, we can associate any generic CQ $q_2(\overline{x}_2)$ with a polynomial $\mathsf{P}^{q_2(\overline{x}_2)}_{q_1(\overline{t})}(\overline{u})$. To do this, first let us observe that the multiplicity of $\overline{t}$ among the answers of $q_2(\overline{x}_2)$ for an arbitrary bag μ over the canonical instance $\mathrm{I}^{q_1(\overline{t})}$ is given by

$$\begin{aligned} q_2^\mu(\overline{t}) &= \sum_{\mathsf{h} \in \mathrm{Hom}(q_2(\overline{x}_2), \mathrm{I}^{q_1(\overline{t})})}^{\mathsf{h}(\overline{x}_2)=\overline{t}} \prod_{\alpha \in \mathrm{body}(\mathsf{h}(q_2(\overline{x})))} \mu(\alpha)^{\mu_{\mathsf{h}(q_2(\overline{x}))}(\alpha)} \\ &= \sum_{\mathsf{h} \in \mathrm{CM}(q_2(\overline{x}_2), q_1(\overline{x}_1))}^{\mathsf{h}(\overline{x}_1)=\overline{t}} \prod_{\alpha \in \mathrm{body}(\mathsf{h}(q_2(\overline{x})))} \mu(\alpha)^{\mu_{\mathsf{h}(q_2(\overline{x}))}(\alpha)} \\ &= \sum_{\mathsf{h} \in \mathrm{CM}(q_2(\overline{x}_2), q_1(\overline{t}))} \prod_{\alpha \in \mathrm{body}(\mathsf{h}(q_2(\overline{x})))} \mu(\alpha)^{\mu_{\mathsf{h}(q_2(\overline{x}))}(\alpha)}. \end{aligned}$$

The following definition becomes immediately clear, once one observes that, for any containment mapping $\mathsf{h} \in \mathrm{CM}(q_2(\overline{x}_2), q_1(\overline{t}))$, the monomial

$$\prod_{\alpha \in \mathrm{body}(\mathsf{h}(q_2(\overline{x})))} \mu(\alpha)^{\mu_{\mathsf{h}(q_2(\overline{x}))}(\alpha)}$$

has the same form of the multiplicity function for the Boolean, projection-free CQ $\mathsf{h}(q_2(\overline{x}))$.

Definition 3.3. *Let* $q_1(\overline{x}_1)$ *be a projection-free* CQ, $\overline{t} \in \mathrm{prbtup}(q_1(\overline{x}_1))$ *one of its probe tuples, and* $\overline{u} \in \mathrm{U}^k$ *a vector of unknowns compatible with* $q_1(\overline{x}_1)$, *for some* $k \in \mathbb{N}$. *Every* CQ $q_2(\overline{x}_2)$ *is associated with the polynomial*

$$\mathsf{P}^{q_2(\overline{x}_2)}_{q_1(\overline{t})}(\overline{u}) \triangleq \sum_{\mathsf{h} \in \mathrm{CM}(q_2(\overline{x}_2), q_1(\overline{t}))} \mathsf{M}_{\mathsf{h}(q_2(\overline{x}_2))}(\overline{u}),$$

w.r.t. $q_1(\overline{x}_1)$ *and* $\overline{t}$.

Notice that, for a given containment mapping from h from $q_2(\overline{x}_2)$ to $q_1(\overline{t})$, it holds that $\mathsf{h}(q_2(\overline{x}_2))$ is a subquery of $q_1(\overline{t})$, thus, the vector of unknowns $\overline{u}$ for $q_1(\overline{x}_1)$ is necessarily compatible with query $\mathsf{h}(q_2(\overline{x}_2))$. This ensures the monomial

$\mathsf{M}_{\mathsf{h}(q_2(\overline{x}_2))}(\overline{u})$ to be well-defined. Moreover, the multiplicity of $\overline{t}$ obtained just by such a homomorphism h on a bag μ over $\mathrm{I}^{q_1(\overline{t}_1)}$, *i.e.*, $(\mathsf{h}(q_2(\overline{x}_2)))^{\mu}(\overline{t})$, is obtained by taking the associated monomial for $\mathsf{h}(q_2(\overline{x}_2))$ and setting its unknowns to the corresponding values of μ for all the atoms in the query. Concluding, the associated polynomial for $q_2(\overline{x}_2)$ *w.r.t.* $q_1(\overline{x}_1)$ and $\overline{t}$, which sums all associated monomials derived from the mappings from $q_2(\overline{x}_2)$ to $q_1(\overline{t})$, gives us the multiplicity of $\overline{t}$ over a bag μ, by simply replacing each unknown u_i in $\overline{u}$ with the multiplicity $\mu(\alpha)$ for the atom α for which $u_i = u^{\alpha}$.

Before proceeding, consider again the projection-free CQ $q_1(x_1, x_2)$ of the previous example, together with its probe tuple $\widehat{x}_1\widehat{x}_2$, and the following additional CQ:

$$q_2(x_1, x_2) \leftarrow R^3(x_1, x_2), R^2(x_1, y_1), R^2(y_2, y_1).$$

Then, it is not hard to see that

$$\mathsf{P}^{q_2(x_1,x_2)}_{q_1(\widehat{x}_1,\widehat{x}_2)}(\overline{u}) = u_1^7 + u_1^5 \cdot u_2^2 + u_1^3 \cdot u_3^4.$$

Indeed, there are only the following three containment mappings from $q_2(x_1, x_2)$ to $q_1(\widehat{x}_1, \widehat{x}_2)$:

$$\mathsf{h}_1 = \{x_1, y_2 \mapsto \widehat{x}_1; x_2, y_1 \mapsto \widehat{x}_2\};$$
$$\mathsf{h}_2 = \{x_1 \mapsto \widehat{x}_1; x_2, y_1 \mapsto \widehat{x}_2; y_2 \mapsto c_1\};$$
$$\mathsf{h}_3 = \{x_1, y_2 \mapsto \widehat{x}_1; x_2 \mapsto \widehat{x}_2; y_1 \mapsto c_2\}.$$

Consequently, the queries $\mathsf{h}_i(q_2(x_1, x_2))$ obtainable from the homomorphisms h_i are:

$$\mathsf{body}(\mathsf{h}_1(q_2(x_1, x_2))) = \{R^7(\widehat{x}_1, \widehat{x}_2)\};$$
$$\mathsf{body}(\mathsf{h}_2(q_2(x_1, x_2))) = \{R^5(\widehat{x}_1, \widehat{x}_2), R^2(c_1, \widehat{x}_2)\};$$
$$\mathsf{body}(\mathsf{h}_3(q_2(x_1, x_2))) = \{R^3(\widehat{x}_1, \widehat{x}_2), R^4(\widehat{x}_1, c_2)\}.$$

Thus, the three monomials $\mathsf{M}_{\mathsf{h}_i(q_2(x_1,x_2))}(\overline{u})$ reported below can be derived, and the previous polynomial $\mathsf{P}^{q_2(x_1,x_2)}_{q_1(\widehat{x}_1,\widehat{x}_2)}(\overline{u})$ immediately follows:

$$\mathsf{M}_{\mathsf{h}_1(q_2(x_1,x_2))}(\overline{u}) = u_1^7;$$
$$\mathsf{M}_{\mathsf{h}_2(q_2(x_1,x_2))}(\overline{u}) = u_1^5 \cdot u_2^2;$$
$$\mathsf{M}_{\mathsf{h}_3(q_2(x_1,x_2))}(\overline{u}) = u_1^3 \cdot u_3^4.$$

We conclude this section by stating the following easy corollary of Theorem 3.1, which characterizes a given bag-containment problem of a projection-free CQ $q_1(\overline{x}_1)$ into a generic CQ $q_2(\overline{x}_2)$ in terms of a Diophantine inequality between the associated monomial $\mathsf{M}_{q_1(\overline{t})}(\overline{u})$ and polynomial $\mathsf{P}^{q_2(\overline{x}_2)}_{q_1(\overline{t})}(\overline{u})$. Note that, in the following, we use $\mathsf{P}(\overline{\xi}) \triangleq \mathsf{P}(\overline{u})|_{\overline{u}=\overline{\xi}}$ to denote the replacement of an unknown u_i in $\overline{u}$ with the value ξ_i in $\overline{\xi}$, for any polynomial P. Moreover, we say that an inequality $\mathsf{M}(\overline{u}) > \mathsf{P}(\overline{u})$ admits a Diophantine solution if there exists a natural vector $\overline{\xi} \in \mathbb{N}^{|\overline{u}|}$ such that $\mathsf{M}(\overline{\xi}) > \mathsf{P}(\overline{\xi})$.

Corollary 3.1. *Given a projection-free* CQ $q_1(\overline{x}_1)$ *and a* CQ $q_2(\overline{x}_2)$*, it holds that* $q_1(\overline{x}_1) \sqsubseteq_b q_2(\overline{x}_2)$ iff*, for all probe tuple* $\overline{t} \in \mathsf{prbtup}(q_1(\overline{x}_1))$*, (1)* $\overline{t}$ *is unifiable with* $\overline{x}_2$ *and (2) the inequality* $\mathsf{M}_{q_1(\overline{t})}(\overline{u}) > \mathsf{P}^{q_2(\overline{x}_2)}_{q_1(\overline{t})}(\overline{u})$ *does not admit any Diophantine solution.*

Note that in Theorem 5.3, we prove that in order to decide bag containment we actually need only one inequality between a monomial and a polynomial. Specifically, we shall just consider the *most-general* probe tuple which we define as the tuple of canonical constants that corresponds to the tuple of free variables of the containee query. However, to prove Theorem 5.3, we first need to use all probe tuples and Corollary 3.1 in order to develop the machinery for such a proof. Using a single probe tuple would obviously make a difference in an implementation of our decision method, but it does not change the asymptotic complexity of the problem.

4 DECIDING INEQUALITIES

We now investigate the problem of deciding whether a Diophantine inequality of a certain structure has an all-natural, *i.e.*, Diophantine, solution. Our inequalities, called *Monomial-Polynomial Inequalities* (MPIs, for short), compare a polynomial on n unknowns, with positive real coefficients, and natural exponents, to a monomial over the same unknowns with coefficient 1 and natural exponents. Although it might be quite evident from the previous sections how a decision procedure for MPIs can help us decide bag containment, the formal statement of the reduction between the two problems is deferred until Section 5. Here we focus on proving decidability for the aforementioned restriction of the generally undecidable *Diophantine inequality problem* [20]. We also discuss another variant of these inequalities, which we call *Generalized Monomial-Polynomial Inequalities* (GMPIs, for short), that differ from the MPIs in that they allow for real exponents.

Observe that, in the remaining part of this work, given two vectors $\overline{u}$ and $\overline{e}$ of the same dimension, we succinctly denote by $\overline{u}^{\overline{e}}$ the monomial $\prod_i {u_i}^{e_i}$. Moreover, recall that the degree of a polynomial is the maximal sum of the exponents over all its monomials.

Definition 4.1. *Given a* n*-dimensional unknown vector* $\overline{u} \in \mathrm{U}^n$*,* m *non-negative real values* $a_1, \ldots, a_m \in \mathbb{R}_{\geq 0}$*, and* $(m+1)$ n*-dimensional integer vectors* $\overline{e}, \overline{e}_1, \ldots, \overline{e}_m \in \mathbb{N}^n$*, with* $m, n \in \mathbb{N}_+$*, the* n-dimensional Monomial-Polynomial Inequality *(*n-MPI*, for short) with coefficients* $a_1, \ldots, a_m$ *and exponents* $\overline{e}, \overline{e}_1, \ldots \overline{e}_m$ *is the syntactic expression reported in the following:*

$$\mathsf{P}(\overline{u}) < \mathsf{M}(\overline{u}), \textit{ with } \mathsf{P}(\overline{u}) \triangleq \sum_{i=1}^{m} a_i \overline{u}^{\overline{e}_i} \textit{ and } \mathsf{M}(\overline{u}) \triangleq \overline{u}^{\overline{e}}. \quad (3)$$

A solution for 3 is a n-dimensional integer vector $\overline{\xi} \in \mathbb{N}^n$ *for which the numeric inequality* $\mathsf{P}(\overline{\xi}) < \mathsf{M}(\overline{\xi})$ *holds. Finally, a* n-dimensional Generalized Monomial-Polynomial Inequality *(*n-GMPI*, for short) is the expression 3, where the exponents* $\overline{e}, \overline{e}_1, \ldots, \overline{e}_m \in \mathbb{R}^n_{\geq 0}$ *are allowed to be positive real vectors.*

Our proof approach has two steps. First, we show that to decide whether a 1-GMPI admits a Diophantine solution, it is enough to compare the degrees between the two sides of the inequality. Then, we prove that for every solvable MPI over n unknowns, there exists a specific 1-GMPI admitting a solution as well. More specifically, we construct an inequality similar to a 1-GMPI, but containing unknowns in the exponents. To compare the degrees of this inequality we then need to solve a homogeneous linear inequality system and show how a Diophantine solution for such a system corresponds to a solution of our original MPI.

To begin, observe that 0 cannot be part of any solution of a n-GMPI $\mathsf{P}(\overline{u}) < \mathsf{M}(\overline{u})$, while 1 cannot be the only value of such a solution, for any $n \in \mathbb{N}_+$ of the unknown vector $\overline{u} \in \mathrm{U}^n$. As an example, consider the 3-MPI

$$u_1^7 + u_1^5 \cdot u_2^2 + u_1^3 \cdot u_3^4 < u_1^2 \cdot u_2 \cdot u_3^3$$

derived at the end of the previous section. Observe that there is no solution with $u_1 = 0$, independently of the value of the other two unknowns u_2 and u_3, since

$$(u_1^7 + u_1^5 \cdot u_2^2 + u_1^3 \cdot u_3^4)|_{u_1=0} = 0 \not< 0 = (u_1^2 \cdot u_2 \cdot u_3^3)|_{u_1=0}.$$

Obviously, the same holds for $u_2 = 0$ or $u_3 = 0$. For similar reasons, $u_1 = u_2 = u_3 = 1$ does not satisfy the 3-MPI under analysis, as

$$(u_1^7 + u_1^5 \cdot u_2^2 + u_1^3 \cdot u_3^4)|_{\substack{u_1=1\\u_2=1\\u_3=1}} = 3 \not< 1 = (u_1^2 \cdot u_2 \cdot u_3^3)|_{\substack{u_1=1\\u_2=1\\u_3=1}}.$$

$$\text{On the contrary,} \quad u_1 = 1, u_2 = 4, \text{ and } u_3 = 3$$

is one of infinitely many Diophantine solutions of the above 3-MPI, since

$$(u_1^7 + u_1^5 \cdot u_2^2 + u_1^3 \cdot u_3^4)|_{\substack{u_1=1\\u_2=4\\u_3=3}} = 98 < 108 = (u_1^2 \cdot u_2 \cdot u_3^3)|_{\substack{u_1=1\\u_2=4\\u_3=3}}.$$

Another solution is, for instance,

$$u_1 = 1, u_2 = 9, \text{ and } u_3 = 3.$$

The following proposition generalizes the previous observation to arbitrary GMPIs.

Proposition 4.1. *Let* $\overline{\xi} \in \mathbb{N}^n$ *be a Diophantine solution of an* n-GMPI $\mathsf{P}(\overline{u}) < \mathsf{M}(\overline{u})$*, with* $n \in \mathbb{N}_+$*. Then,* $\xi_j \geq 1$*, for all* $j \in [1, n]$*, and* $\xi_j > 1$*, for some* $j \in [1, n]$*.*

Proof. First notice that, if $\xi_j = 0$, for some $j \in [1, n]$, we would have $\mathsf{M}(\overline{\xi}) = 0$, thus, $0 \leq \mathsf{P}(\overline{\xi}) < 0 = \mathsf{M}(\overline{\xi})$, which is impossible. Now, suppose that $\xi_j = 1$, for all $j \in [1, n]$. Then, $\mathsf{M}(\overline{\xi}) = 1$, but $1 \leq \mathsf{P}(\overline{\xi})$, hence, $1 \leq \mathsf{P}(\overline{\xi}) < 1 = \mathsf{M}(\overline{\xi})$, which is impossible as well. □

As one might observe from the previous discussion, the relationship between the degree of the polynomial $\mathsf{P}(\overline{u})$ and the monomial $\mathsf{M}(\overline{u})$ in a n-GMPI does not constitute, in general, a criterion to decide whether this inequality admits a Diophantine solution. If $\mathsf{P}(u)$ has strictly lower degree than $\mathsf{M}(u)$, indeed, it can be shown that $\mathsf{P}(u) < \mathsf{M}(u)$ is necessarily solvable. However, a solution may even exist if the opposite holds, as exemplified by the above 3-MPI, where we have

$$\deg\left(u_1^7 + u_1^5 \cdot u_2^2 + u_1^3 \cdot u_3^4\right) = 7 > 6 = \deg\left(u_1^2 \cdot u_2 \cdot u_3^3\right).$$

In case of dimension 1, however, a 1-GMPI $\mathsf{P}(u) < \mathsf{M}(u)$ is solvable precisely when the degree of $\mathsf{P}(u)$ is strictly lower than the degree of $\mathsf{M}(u)$. For instance, $u^4 + u^2 < u^4$ is unsolvable, since $\deg\left(u^4 + u^2\right) = 4 \not< 4 = \deg\left(u^4\right)$. On the contrary, 3 is a solution for $2 \cdot u^4 + 1 < u^5$. Intuitively, a 1-GMPI is solvable whenever the monomial has an asymptotic grow that is greater than the one of the polynomial, so that there is necessarily a point after which $\mathsf{M}(\overline{u})$ assumes values greater than those assumed by $\mathsf{P}(\overline{u})$. The following lemma formalizes this concept.

Lemma 4.1. *A* 1-GMPI $\mathsf{P}(u) < \mathsf{M}(u)$ *admits a positive Diophantine solution* iff $\deg(\mathsf{P}(u)) < \deg(\mathsf{M}(u))$.

Proof. **(If).** Suppose that $\deg(\mathsf{P}(u)) < \deg(\mathsf{M}(u))$. Then, $e_i < e$, for all $i \in [1, m]$, where $e_i \in \mathbb{R}_+$ is the i-th exponent of the polynomial $\mathsf{P}(u) = \sum_{i=1}^m a_i u^{e_i}$ and $e \in \mathbb{R}_+$ is the exponent of the monomial $\mathsf{M}(u) = u^e$. Due to the well known special limit of rational functions, it holds that

$$\lim_{u \to +\infty} a_i u^{e_i} / u^e = 0,$$

where $a_i \in \mathbb{R}_+$ is the coefficient of the i-th monomial in the polynomial $\mathsf{P}(u)$. Therefore, by expanding the definition of limit, we have that there exists a positive real number $\ell_i \in \mathbb{R}_+$ such that

$$a_i u^{e_i} / u^e < 1/m,$$

for all $u \in \mathbb{R}_+$ with $u > \ell_i$. Now, assume

$$\xi^\star \triangleq 1 + \left\lceil \max_{i \in [1,m]} \ell_i \right\rceil \in \mathbb{N}_+.$$

By construction,

$$a_i \xi^{\star e_i} / \xi^{\star e} < 1/m, \text{ i.e., } a_i \xi^{\star e_i} < \xi^{\star e}/m,$$

for all $i \in [1, m]$. Consequently,

$$\mathsf{P}(\xi^\star) = \sum_{i=1}^m a_i \xi^{\star e_i} < \sum_{i=1}^m \xi^{\star e}/m = \xi^{\star e} = \mathsf{M}(\xi^\star).$$

So, $\xi^\star$ is a positive Diophantine solution for $\mathsf{P}(u) < \mathsf{M}(u)$.

(Only if). Suppose that $\deg(\mathsf{P}(u)) \geq \deg(\mathsf{M}(u))$. Then, there exists an index $i \in [1, m]$ such that $e_i \geq e$. Since $a_i \geq 1$, we have that

$$a_i u^{e_i}/u^e \geq u^{e_i}/u^e = u^{e_i - e}.$$

Moreover, $u^{e_i - e} \geq 1$, for all $u \in \mathbb{R}_+$ with $u \geq 1$, being $e_i - e$ non negative, which implies $a_i u^{e_i} \geq u^e$. Thus,

$$\mathsf{P}(u) = \sum_{i=1}^{m} a_i u^{e_i} \geq u^e = \mathsf{M}(u),$$

for all $u \in \mathbb{N}_+$. Consequently, $\mathsf{P}(u) < \mathsf{M}(u)$ cannot admit any positive Diophantine solution. □

In the multi-dimensional case, the reason why there is no similar simple dependence between the existence of a Diophantine solution for an GMPI $\mathsf{P}(\overline{u}) < \mathsf{M}(\overline{u})$ and the relationship between the degrees of its parts is that different unknowns may have different degrees in their contribution to the value of the polynomial $\mathsf{P}(\overline{u})$ and monomial $\mathsf{M}(\overline{u})$. For instance, in the search for a solution of the 3-MPI previously analyzed, it is counterproductive to have values for u_1 greater than 1, since this unknown appears in all monomials of

$$\mathsf{P}(\overline{u}) = u_1^7 + u_1^5 \cdot u_2^2 + u_1^3 \cdot u_3^4$$

with greater degree than the one in

$$\mathsf{M}(\overline{u}) = u_1^2 \cdot u_2 \cdot u_3^3$$

and thus it contributes more to value of the former than to that of the latter. On the contrary, u_2 and u_3 are completely decoupled in $\mathsf{P}(\overline{u})$ *w.r.t.* the operation of multiplication, since the first one only appears squared in the second monomial, while the second one just occurs in the third monomial as a fourth power. In other words, their combined contribution to $\mathsf{P}(\overline{u})$ is additive, while in $\mathsf{M}(\overline{u})$ is multiplicative. Consequently, by fixing $u_1 = 1$ and suitably choosing values for u_2 and u_3, we might be able to ensure that the sum

$$\mathsf{P}(\overline{u}) = 1 + u_2^2 + u_3^4$$

assumes smaller values than the product

$$\mathsf{M}(\overline{u}) = u_2 \cdot u_3^3,$$

if a certain linear relationship holds between the exponents as we show below (we already know that such values exist: $u_2 = 4$ and $u_3 = 3$). In order to find these values, we can express all unknowns as possibly different powers of a fresh unknown u transforming the 3-MPI

$$u_1^7 + u_1^5 \cdot u_2^2 + u_1^3 \cdot u_3^4 < u_1^2 \cdot u_2 \cdot u_3^3$$

into a parametric 1-MPI $\mathsf{P}^\star(u) < \mathsf{M}^\star(u)$, with

$$\begin{aligned}\mathsf{P}^\star(u) &\triangleq u^{7\epsilon_1} + u^{5\epsilon_1} \cdot u^{2\epsilon_2} + u^{3\epsilon_1} \cdot u^{4\epsilon_3} \\ &= u^{7\epsilon_1} + u^{5\epsilon_1 + 2\epsilon_2} + u^{3\epsilon_1 + 4\epsilon_3}, \\ \mathsf{M}^\star(u) &\triangleq u^{2\epsilon_1} \cdot u^{\epsilon_2} \cdot u^{3\epsilon_3} = u^{2\epsilon_1 + \epsilon_2 + 3\epsilon_3},\end{aligned}$$

such that the first inequality is solvable iff the second one is solvable as well, for some choice of the parameter vector $\overline{\epsilon} = (\epsilon_1, \epsilon_2, \epsilon_3)$. It is immediate to prove, indeed, that the scalar $\xi^\star \in \mathbb{N}_+$ is a solution for $\mathsf{P}^\star(u) < \mathsf{M}^\star(u)$ *iff* the vector $\overline{\xi} \in \mathbb{N}_+^3$ is a solution for $\mathsf{P}(\overline{u}) < \mathsf{M}(\overline{u})$, where $\epsilon_j \triangleq \log_{\xi^\star}(\xi_j)$ At this point, by exploiting the above lemma, we can verify the existence of a solution for $\mathsf{P}^\star(u) < \mathsf{M}^\star(u)$, by checking the existence of a real vector $\overline{\epsilon} \in \mathbb{R}^3$ for which

$$\begin{aligned}\deg\big(\mathsf{P}^\star(u)\big) &= \max(7\epsilon_1, 5\epsilon_1 + 2\epsilon_2, 3\epsilon_1 + 4\epsilon_3) \\ &< 2\epsilon_1 + \epsilon_2 + 3\epsilon_3 \\ &= \deg\big(\mathsf{M}^\star(u)\big)\end{aligned}$$

holds. The latter is then equivalent to the homogeneous linear inequality system

$$\begin{aligned}7\epsilon_1 &< 2\epsilon_1 + \epsilon_2 + 3\epsilon_3 \\ 5\epsilon_1 + 2\epsilon_2 &< 2\epsilon_1 + \epsilon_2 + 3\epsilon_3 \\ 3\epsilon_1 + 4\epsilon_3 &< 2\epsilon_1 + \epsilon_2 + 3\epsilon_3,\end{aligned}$$

which can be simplified into

$$\begin{aligned}-5\epsilon_1 + \epsilon_2 + 3\epsilon_3 &> 0 \\ -3\epsilon_1 - \epsilon_2 + 3\epsilon_3 &> 0 \\ -\epsilon_1 - \epsilon_2 + 3\epsilon_3 &> 0.\end{aligned}$$

A possible solution for this system is $\epsilon_1 = 0$, $\epsilon_2 = 2$, and $\epsilon_3 = 1$, from which we derive the 1-MPI

$$2 \cdot u^4 + 1 < u^5$$

having 3 as one of its infinite solutions. We finally compute the Diophantine solution $\xi_1 = 1, \xi_2 = 9, \xi_3 = 3$ for the 3-MPI under analysis by using the formula $\xi_j = \xi^{\star \epsilon_j}$. We can now state the general criterion. Observe that here we only present the result for MPIs, since this suffices for our purposes, and leave the more general version of the theorem as future work.

Theorem 4.1. *An n-*MPI $\mathsf{P}(\overline{u}) < \mathsf{M}(\overline{u})$*, with $n \in \mathbb{N}_+$, admits a positive Diophantine solution* iff *the system of n-dimensional homogeneous linear inequalities $\{(\overline{e} - \overline{e}_i)^\intercal \cdot \overline{\epsilon} > 0\}_{i=1}^m$ admits a Diophantine solution as well, where $\overline{e}, \overline{e}_1, \ldots \overline{e}_m \in \mathbb{N}^n$ are the exponents used in the n-*MPI *itself, with $m \in \mathbb{N}_+$.*

Proof. **(If).** Suppose that the system

$$\{(\overline{e} - \overline{e}_i)^\intercal \cdot \overline{\epsilon} > 0\}_{i=1}^m$$

admits a Diophantine solution $\overline{d} \in \mathbb{N}^n$ and consider the 1-MPI $\mathsf{P}^\star(u) < \mathsf{M}^\star(u)$ with

$$\mathsf{P}^\star(u) \triangleq \sum_{i=1}^{m} a_i u^{(\overline{e}_i{}^\intercal \cdot \overline{d})} \text{ and } \mathsf{M}^\star(u) \triangleq u^{(\overline{e}^\intercal \cdot \overline{d})},$$

where each $a_i \in \mathbb{R}_{\geq 1}$ is the i-th coefficient of the polynomial $\mathsf{P}(\overline{u})$. By construction, it holds that

$$\deg\big(\mathsf{P}^\star(u)\big) < \deg\big(\mathsf{M}^\star(u)\big),$$

since $(\overline{e}^\intercal \cdot \overline{d}) - (\overline{e}_i{}^\intercal \cdot \overline{d}) = (\overline{e} - \overline{e}_i)^\intercal \cdot \overline{d} > 0$, for all $i \in [1, m]$. Thus, by Lemma 4.1, we have $\mathsf{P}^\star(u) < \mathsf{M}^\star(u)$ admits a positive Diophantine solution $\xi^\star \in \mathbb{N}_+$. Now consider the n-dimensional natural vector $\overline{\xi} \in \mathbb{N}^n$ whose components are defined as $\xi_j \triangleq \xi^{\star d_j}$, for all $j \in [1, n]$. It is immediate to see that $\overline{\xi}$ is a Diophantine solution for the original n-MPI $\mathsf{P}(\overline{u}) < \mathsf{M}(\overline{u})$, since

$$\mathsf{P}(\overline{\xi}) = \mathsf{P}^\star(\xi^\star) < \mathsf{M}^\star(\xi^\star) = \mathsf{M}(\overline{\xi}).$$

Indeed,

$$\mathsf{M}(\overline{\xi}) = \overline{\xi}^{\overline{e}} = \prod_{j=1}^{n} \xi_j{}^{e_j} = \prod_{j=1}^{n} \xi^{\star d_j e_j} = \xi^{\star \sum_{j=1}^{n} e_j d_j} = \xi^{\star (\overline{e}^\intercal \cdot \overline{d})} = \mathsf{M}^\star(\xi^\star).$$

The same reasoning applies to each monomial of the polynomial $\mathsf{P}(\overline{u})$ and, so, to the polynomial itself.

(Only if). Suppose that $\mathsf{P}(\overline{u}) < \mathsf{M}(\overline{u})$ admits a Diophantine solution $\overline{\xi} \in \mathbb{N}^n$ and let $\xi^\star \triangleq \min_{j \in [1,n]}^{\xi_j > 1} \xi_j$. Observe that the existence of such a positive integer number $\xi^\star$ is ensured by Proposition 4.1. Now, let $\overline{r} \in \mathbb{R}^n_{\geq 0}$ be the n-dimensional real vector having components defined as $r_j \triangleq \log_{\xi^\star}(\xi_j)$, for all $j \in [1, n]$, whose existence is ensured again by Proposition 4.1, since $\xi_j \geq 1$. Obviously, $\xi_j = \xi^{\star r_j}$. Moreover, consider the 1-GMPI $\mathsf{P}^\star(u) < \mathsf{M}^\star(u)$ with

$$\mathsf{P}^\star(u) \triangleq \sum_{i=1}^{m} a_i u^{(\overline{e}_i{}^\intercal \cdot \overline{r})} \text{ and } \mathsf{M}^\star(u) \triangleq u^{(\overline{e}^\intercal \cdot \overline{r})},$$

where the coefficients a_i are the same as in the previous case. It is immediate to see that $\xi^\star$ is a positive Diophantine solution for $\mathsf{P}^\star(u) < \mathsf{M}^\star(u)$, since

$$\mathsf{P}^\star(\xi^\star) = \mathsf{P}(\overline{\xi}) < \mathsf{M}(\overline{\xi}) = \mathsf{M}^\star(\xi^\star).$$

Indeed,

$$\mathsf{M}(\overline{\xi}) = \overline{\xi}^{\overline{e}} = \prod_{j=1}^{n} \xi_j{}^{e_j} = \prod_{j=1}^{n} \xi^{\star r_j e_j} = \xi^{\star \sum_{j=1}^{n} e_j r_j} = \xi^{\star (\overline{e}^\intercal \cdot \overline{r})} = \mathsf{M}^\star(\xi^\star).$$

A similar reasoning applies to the polynomial $\mathsf{P}(\overline{u})$. At this point, by Lemma 4.1, we have that

$$\deg\big(\mathsf{P}^\star(u)\big) < \deg\big(\mathsf{M}^\star(u)\big),$$

which means that

$$\overline{e}_i{}^\intercal \cdot \overline{r} < \overline{e}^\intercal \cdot \overline{r}$$

and, so, $(\overline{e} - \overline{e}_i)^\intercal \cdot \overline{r} > 0$, for all $i \in [1, m]$. Hence, $\overline{r}$ is a solution in $\mathbb{R}^n_{\geq 0}$ for the linear system

$$\{(\overline{e} - \overline{e}_i)^\intercal \cdot \overline{\epsilon} > 0\}_{i=1}^{m}.$$

Now, due to the fact that the coefficients $\overline{e} - \overline{e}_i$ are all rational, there is necessarily an n-dimensional rational solution $\overline{q} \in \mathbb{Q}^n_{\geq 0}$ for $\{(\overline{e} - \overline{e}_i)^\intercal \cdot \overline{\epsilon} > 0\}_{i=1}^{m}$ [29]. Recall that, for every $j \in [1, n]$, there exist two natural numbers $a_i \in \mathbb{N}$ and $b_j \in \mathbb{N}_+$ with $\gcd(a_i, b_j) = 1$ such that we can write the j^{th} component of $\overline{q}$ as $q_j = a_i / b_j$ and consider the n-dimensional natural vector $\overline{d} \in \mathbb{N}^n$ whose components are defined as follows: $d_j = bq_j$, for all $j \in [1, n]$, where $b \triangleq \mathrm{lcm}_{j=1}^{n}(b_j) \geq 1$. Then, it is evident that $\overline{d}$ is a Diophantine solution for the system as well, as $(\overline{e} - \overline{e}_i)^\intercal \cdot \overline{d} = b(\overline{e} - \overline{e}_i)^\intercal \cdot \overline{q} > 0$. □

At this point, it is not hard to prove that the Diophantine solution problem for a given n-MPI can be solved in PTime by reducing it to the feasibility problem of the associated homogeneous linear inequality system.

Theorem 4.2. *The Diophantine solution problem for an n-MPI $\mathsf{P}(\overline{u}) < \mathsf{M}(\overline{u})$, with $n \in \mathbb{N}_+$, is in PTime.*

Proof. Consider an n-MPI $\mathsf{P}(\overline{u}) < \mathsf{M}(\overline{u})$ defined over $(m + 1)$ exponents $\overline{e}, \overline{e}_1, \ldots \overline{e}_m \in \mathbb{N}_n$, with $n, m \in \mathbb{N}_+$. By Theorem 4.1, it holds that $\mathsf{P}(\overline{u}) < \mathsf{M}(\overline{u})$ admits a Diophantine solution *iff* the system of n-dimensional homogeneous linear inequalities $\{(\overline{e} - \overline{e}_i)^\intercal \cdot \overline{\epsilon} > 0\}_{i=1}^{m}$ admits a Diophantine solution as well. As already observed in the proof of the above theorem, the latter question is equivalent to the classic rational feasibility problem of linear systems, which is solvable in polynomial time *w.r.t.* its size [29], *i.e.*, in the dimension n (unknowns in $\mathsf{M}(\overline{u})$), number of constraints (monomials in $\mathsf{P}(\overline{u})$), and value of the coefficients (exponents $\overline{e}, \overline{e}_1, \ldots \overline{e}_m$). Hence, the thesis immediately follows. □

5 COMPLEXITY RESULTS

By exploiting the decidability of the restricted version of the Diophantine inequality problem previously discussed, we are now able to decide the bag-containment problem of a projection-free CQ into a generic one [2].

A naive reduction from the problem we are interested in to the associated linear inequality problem, via Corollary 3.1 and Theorems 4.1 and 4.2, can be described as follows. First, the corollary allows to reduce a negative instance

$$q_1(\overline{x}_1) \not\subseteq_{\mathsf{b}} q_2(\overline{x}_2)$$

of the bag-containment problem to the solution of at least one among a set of MPI problems

$$\mathsf{P}^{q_2(\overline{x}_2)}_{q_1(\overline{t})}(\overline{u}) < \mathsf{M}_{q_1(\overline{t})}(\overline{u}),$$

each one for a different probe tuple $\overline{t} \in \mathsf{prbtup}(q_1(\overline{x}_1))$. Then, thanks to Theorem 4.1, the search of such a solution can be in its turn reduced to verify the existence of a solution for the associated inequality system

$$\{(\overline{e} - \overline{e}_{\mathsf{h}})^\intercal \cdot \overline{\epsilon} > 0\}_{\mathsf{h} \in \mathrm{CM}(q_2(\overline{x}_2), q_1(\overline{t}))},$$

where $\overline{e}$ and $\overline{e}_{\mathsf{h}}$ are the exponent vectors of the monomials $\mathsf{M}_{q_1(\overline{t})}(\overline{u})$ and $\mathsf{M}_{\mathsf{h}(q_2(\overline{x}_2))}(\overline{u})$, respectively, the latter being contained into the polynomial $\mathsf{P}^{q_2(\overline{x}_2)}_{q_1(\overline{t})}(\overline{u})$. Unfortunately, since

there might be exponentially many containment mappings $\mathsf{h} \in \mathrm{CM}(q_2(\overline{x}_2), q_1(\overline{t}))$, this approach would let us obtain an ExpSpace decidability procedure only. Indeed, the size of the polynomial associated with the containee query, and so the number of inequalities in the linear system, is generally exponential in the size of the query itself. In order to avoid such exponential blow up in the required space, we put in practice a guess&check approach, which allows us to describe a procedure that decides containment, whose complexity is Π_2^{P} *w.r.t.* the size of the containing query and CoNPTime *w.r.t.* the size of the containee query. To do this, we make use of the following simple reformulation of a known result [32], which provides a polynomial bound on the size of a possible solution for an integer inequality system.

Lemma 5.1. *Let $\overline{e}_1, \ldots, \overline{e}_m \in \mathbb{Z}^n$ and $\overline{f} \in \mathbb{Z}^m$ be $(m+1)$ integer vectors, with $m, n \in \mathbb{N}_+$. An n-dimensional linear inequality system $\{\overline{e}_i{}^{\intercal} \cdot \overline{v} \leq f_i\}_{i=1}^m$ admits a positive solution* iff *it admits a natural one $\overline{\xi} \in \mathbb{N}^n$ with $\sum_{i=1}^n \xi_i \leq 6n^3\phi$, where $\phi \triangleq \max_{i=1}^m(f_i + \sum_{j=1}^n e_{i,j})$ is the maximum sum of the coefficients in an inequality of the system plus its constant term.*

The idea behind our guess&check approach is quite simple: suitably combine Corollary 3.1, Theorem 4.1, and Lemma 5.1 into a single criterion and then verify whether this holds by using a $\forall\exists$-alternating Turing machine, *i.e.*, equivalently, a CoNPTime Turing machine with an NPTime oracle. In particular, in order to avoid the exponential blow-up in space complexity, we do not write down the entire system, but just guess which inequality might not be satisfied by a given proposed solution. In more detail, we first universally guess a probe tuple $\overline{t} \in \mathrm{prbtup}(q_1(\overline{x}_1))$ (one that would "break containment") and a hypothetical solution $\overline{d} \in \mathbb{N}^{|\overline{e}|}$ of the system $\{(\overline{e} - \overline{e}_{\mathsf{h}})^{\intercal} \cdot \overline{\epsilon} > 0\}_{\mathsf{h} \in \mathrm{CM}(q_2(\overline{x}_2), q_1(\overline{t}))}$ and then verify it by existentially guessing which inequality $(\overline{e} - \overline{e}_{\mathsf{h}})^{\intercal} \cdot \overline{\epsilon} > 0$ might be violated. Thanks to Lemma 5.1, the existence of a polynomial bound $\mathrm{sb}(q_1(\overline{t}), q_2(\overline{x}_2)) \triangleq \max_{\mathsf{h} \in \mathrm{CM}(q_2(\overline{x}_2), q_1(\overline{t}))} \sum_j (\overline{e} - \overline{e}_{\mathsf{h}})_j$ on a possible solution of the system ensures that the guessing has polynomial size. Note that in the inequalities of Theorem 4.1 there is no constant term (as f_i in Lemma 5.1). Next theorem formalizes the discussed criterion.

Theorem 5.1. *Let $q_1(\overline{x}_1)$ and $q_2(\overline{x}_2)$ be two* CQ*s, the first of which is projection-free. Then, $q_1(\overline{x}_1) \sqsubseteq_{\mathsf{b}} q_2(\overline{x}_2)$* iff*, for any probe tuple $\overline{t} \in \mathrm{prbtup}(q_1(\overline{x}_1))$, it holds that (i) $\overline{t}$ is unifiable with $\overline{x}_2$ and, (ii) assuming $n = \left|\mathrm{body}(q_1(\overline{t}))\right|$, for all natural vectors $\overline{d} \in \mathbb{N}^n$ with $\sum_j d_j \leq \mathrm{sb}(q_1(\overline{t}), q_2(\overline{x}_2))$, there exists a containment mapping $\mathsf{h} \in \mathrm{CM}(q_2(\overline{x}_2), q_1(\overline{t}))$ such that $(\overline{e} - \overline{e}_{\mathsf{h}})^{\intercal} \cdot \overline{d} \leq 0$, where $\overline{e}$ and $\overline{e}_{\mathsf{h}}$ are the exponent vectors of the monomials $\mathsf{M}_{q_1(\overline{t})}(\overline{u})$ and $\mathsf{M}_{\mathsf{h}(q_2(\overline{x}_2))}(\overline{u})$, respectively.*

Proof. By Corollary 3.1, we know that $q_1(\overline{x}_1) \sqsubseteq_{\mathsf{b}} q_2(\overline{x}_2)$ *iff*, for all probe tuples $\overline{t} \in \mathrm{prbtup}(q_1(\overline{x}_1))$, it holds that (i) $\overline{t}$ is unifiable with $\overline{x}_2$ and (ii) the MPI

$$\mathsf{M}_{q_1(\overline{t})}(\overline{u}) > \sum_{\mathsf{h} \in \mathrm{CM}(q_2(\overline{t}), q_1(\overline{t}))} \mathsf{M}_{\mathsf{h}(q_2(\overline{x}_2))}(\overline{u})$$

does not have a Diophantine solution. In addition, due to Theorem 4.1, we derive that the second item is equivalent to the fact that the inequality system

$$\{(\overline{e} - \overline{e}_{\mathsf{h}})^{\intercal} \cdot \overline{v} > 0\}_{\mathsf{h} \in \mathrm{Hom}(q_2(\overline{t}), q_1(\overline{t}))}$$

does not have a Diophantine solution as well. This implies that, by Lemma 5.1, every natural vector $\overline{d} \in \mathbb{N}^n$ with $\sum_j d_j \leq \mathrm{sb}(q_1(\overline{t}), q_2(\overline{x}_2))$ cannot be a solution of the system *iff* this is unsolvable, which means that there exists at least one inequality $(\overline{e} - \overline{e}_{\mathsf{h}})^{\intercal} \cdot \overline{v} > 0$, and so one containment mapping $\mathsf{h} \in \mathrm{CM}(q_2(\overline{x}_2), q_1(\overline{t}))$, such that $(\overline{e} - \overline{e}_{\mathsf{h}})^{\intercal} \cdot \overline{d} \leq 0$ holds. The statement easily follows. □

At this point, we can prove our main result that concerns the upper-bound complexity of the bag-containment problem of a projection-free CQ into a generic one.

Theorem 5.2. *The bag-containment problem $q_1(\overline{x}_1) \sqsubseteq_{\mathsf{b}} q_2(\overline{x}_2)$ of a projection-free* CQ *$q_1(\overline{x}_1)$ into a* CQ *$q_2(\overline{x}_2)$ is solvable in Π_2^{P}* w.r.t. *the size of containing query and in CoNPTime* w.r.t. *the size of containee query.*

Proof. To prove the required statement we describe a polynomial-size $\forall\exists$-alternating procedure that, given the two queries $q_1(\overline{x}_1)$ and $q_2(\overline{x}_2)$ in input, exhibits an accepting computation *iff* $q_1(\overline{x}_1) \sqsubseteq_{\mathsf{b}} q_2(\overline{x}_2)$ holds. We leave to the interesting reader the tedious but trivial translation of our procedure to an equivalent polynomial-size $\forall\exists$-alternating Turing machine $\mathcal{M}$.

The procedure starts in a universal state s_0 by guessing a probe tuple $\overline{t} \in \mathrm{prbtup}(q_1(\overline{x}_1))$. If $\overline{t}$ is not unifiable with $\overline{x}_2$, the computation is rejected and the procedure terminates. Otherwise, a transit to a universal state $s_{\overline{t}}$ occurs, from which a natural vector $\overline{d} \in \mathbb{N}^n$ with $\sum_j d_j \leq \mathrm{sb}(q_1(\overline{t}), q_2(\overline{x}_2))$ and $n = \left|\mathrm{body}(q_1(\overline{t}))\right|$ is chosen. This step allows to reach an existential state $s_{\overline{t},\overline{d}}$. At this point, a containment mapping $\mathsf{h} \in \mathrm{CM}(q_2(\overline{x}_2), q_1(\overline{t}))$ is guessed as well and the procedure transits to the final state $s_{\overline{t},\overline{d},\mathsf{h}}$, which is accepting *iff* the homogeneous linear inequality $(\overline{e} - \overline{e}_{\mathsf{h}})^{\intercal} \cdot \overline{d} \leq 0$ holds, where $\overline{e}$ and $\overline{e}_{\mathsf{h}}$ are the exponent vectors of the monomials $\mathsf{M}_{q_1(\overline{t})}(\overline{u})$ and $\mathsf{M}_{\mathsf{h}(q_2(\overline{x}_2))}(\overline{u})$, respectively. Now, one can make the following easy observations about the guessing points s_0, $s_{\overline{t}}$, $s_{\overline{t},\overline{d}}$ and the final state $s_{\overline{t},\overline{d},\mathsf{h}}$:

(1) the probe tuple $\overline{t} \in \mathrm{prbtup}(q_1(\overline{x}_1))$ requires (in binary encoding) space $\mathrm{O}(m \cdot \log(m + c))$, where m and c are the numbers of variables and constants in $\overline{x}_1$ and $q_1(\overline{x}_1)$, respectively;

(2) the natural vector $\overline{d} \in \mathbb{N}^n$ requires space $O(n \cdot \log(b))$, where the n atoms in $q_1(\overline{t})$ are not more than those in $q_1(\overline{x})$, and $b = \mathsf{sb}(q_1(\overline{t}), q_2(\overline{x}_2))$ is the maximum value an unknown can assume;
(3) the containment mapping $\mathsf{h} \in \mathrm{CM}(q_2(\overline{x}_2), q_1(\overline{t}))$ requires space $O(m \cdot \log(n))$, where n and m are the number of atoms in $q_1(\overline{t})$ and $q_2(\overline{x})$, respectively;
(4) the inequality $(\overline{e} - \overline{e}_{\mathsf{h}})^{\intercal} \cdot \overline{d} \leq 0$ contains n summands and we can verify it in PTime.

Evidently, thanks to the relationship between the alternation of quantification types among the guessing points and the levels of the polynomial hierarchy [4, 6], the procedure has complexity Π_2^{p} *w.r.t.* the size of containing query and in CoNPTime *w.r.t.* the size of containee query [33]. Finally, we can immediately observe that soundness and completeness of the described approach directly from Theorem 5.1. □

Before concluding with a hardness reduction, we provide a stronger result that simplifies our characterization via Diophantine inequalities, namely Corollary 3.1 and Theorem 5.1, without improving though the asymptotic complexity of the decision problem. Specifically, we show that we just need to evaluate the MPI associated with a bag-containment problem for the most-general probe tuple, similarly to what one would do for classic set-containment. The difference here is that, to do so, we cannot work only at the level of database instances and homomorphisms, but need to exploit the solution of an MPI via the corresponding linear system.

Theorem 5.3. *Given a projection-free* CQ $q_1(\overline{x}_1)$ *and a* CQ $q_2(\overline{x}_2)$*, it holds that* $q_1(\overline{x}_1) \sqsubseteq_b q_2(\overline{x}_2)$ iff *the inequality* $\mathsf{M}_{q_1(\overline{t}^\star)}(\overline{u}) > \mathsf{P}^{q_2(\overline{x}_2)}_{q_1(\overline{t}^\star)}(\overline{u})$ *does not admit a Diophantine solution, where* $\overline{t}^\star \in \mathsf{prbtup}(q_1(\overline{x}_1))$ *is the most-general probe tuple.*

Proof. Thanks to Theorem 5.1, we know that

$$q_1(\overline{x}_1) \not\sqsubseteq_b q_2(\overline{x}_2)$$

iff there exists a probe tuple $\overline{t} \in \mathsf{prbtup}(q_1(\overline{x}_1))$ and a natural vector $\overline{d} \in \mathbb{N}^n$, with $n = \left|\mathsf{body}(q_1(\overline{t}))\right|$, such that either (i) $\overline{t}$ is not unifiable with $\overline{x}_2$ or, (ii) for all containment mappings $\mathsf{h} \in \mathrm{CM}(q_2(\overline{x}_2), q_1(\overline{t}))$, it holds that

$$(\overline{e} - \overline{e}_{\mathsf{h}})^{\intercal} \cdot \overline{d} > 0,$$

where $\overline{e}$ and $\overline{e}_{\mathsf{h}}$ are the exponent vectors of the monomials $\mathsf{M}_{q_1(\overline{t})}(\overline{u})$ and $\mathsf{M}_{\mathsf{h}(q_2(\overline{x}_2))}(\overline{u})$, respectively. Consider the most-general probe tuple $\overline{t}^\star \in \mathsf{prbtup}(q_1(\overline{x}_1))$, the monomial $\mathsf{M}_{q_1(\overline{t}^\star)}(\overline{u})$, every monomial $\mathsf{M}_{\mathsf{h}^\star(q_2(\overline{x}_2))}(\overline{u})$ for some given containment mapping $\mathsf{h}^\star \in \mathrm{CM}(q_2(\overline{x}_2), q_1(\overline{t}^\star))$, and the corresponding exponent vectors $\overline{e}^\star$ and $\overline{e}^\star_{\mathsf{h}^\star}$, whose dimension is $n^\star = \left|\mathsf{body}(q_1(\overline{t}^\star))\right| = |\mathsf{body}(q_1(\overline{x}_1))|$. We now show that

$$q_1(\overline{x}_1) \not\sqsubseteq_b q_2(\overline{x}_2)$$

iff there exists a natural vector $\overline{d}^\star \in \mathbb{N}^{n^\star}$ such that

$$(\overline{e}^\star - \overline{e}^\star_{\mathsf{h}^\star})^{\intercal} \cdot \overline{d}^\star > 0,$$

for all containment mappings $\mathsf{h}^\star \in \mathrm{CM}(q_2(\overline{x}_2), q_1(\overline{t}^\star))$. First observe that every containment mapping $\mathsf{h}^\star \in \mathrm{CM}(q_2(\overline{x}_2), q_1(\overline{t}^\star))$ induces a containment mapping $\mathsf{h}^\star \circ \mathsf{f} \in \mathrm{CM}(q_2(\overline{x}_2), q_1(\overline{t}))$, where f is the unique surjective homomorphism from $q_1(\overline{t}^\star)$ to $q_1(\overline{t})$, *i.e.*, more precisely, from $\mathrm{I}^{q_1(\overline{t}^\star)}$ to $\mathrm{I}^{q_1(\overline{t})}$. Also, assume the bodies of the two queries $q_1(\overline{t}^\star)$ and $q_1(\overline{t})$ to be enumerated, *i.e.*,

$$\mathsf{body}(q_1(\overline{t}^\star)) = \{\alpha_1^\star, \ldots, \alpha_{n^*}^\star\}, \mathsf{body}(q_1(\overline{t})) = \{\alpha_1, \ldots, \alpha_n\},$$

and let $\mathrm{g}\colon [1, n^\star] \to [1, n]$ be the unique surjective map between atom indexes induced by the homomorphism f as defined in the following: for all atoms $\alpha_i^\star \in \mathsf{body}(q_1(\overline{t}^\star))$, it holds that $\mathsf{f}(\alpha_i^\star) = \alpha_{\mathrm{g}(i)}$ is the $\mathrm{g}(i)$-th atom in $\mathsf{body}(q_1(\overline{t}))$. Obviously, the enumeration needs to associate every atom $\alpha_i^\star$ and α_j with the corresponding positions $i \in [1, n^\star]$ and $j \in [1, n]$ in all the vectors of the inequalities. At this point, we claim that the natural vector $\overline{d}^\star \in \mathbb{N}^{n^\star}$, with $d_i^\star \triangleq d_{\mathrm{g}(i)}$, for all $i \in [1, n^\star]$, satisfies all the required inequalities $(\overline{e}^\star - \overline{e}^\star_{\mathsf{h}^\star})^{\intercal} \cdot \overline{d}^\star > 0$, with $\mathsf{h}^\star \in \mathrm{CM}(q_2(\overline{x}_2), q_1(\overline{t}^\star))$. To do this, observe that

$$e_j = \sum_{\substack{i=1 \\ \mathrm{g}(i)=j}}^{n^\star} e_i^\star \text{ and } e_{(\mathsf{h}^\star \circ \mathsf{f}),j} = \sum_{\substack{i=1 \\ \mathrm{g}(i)=j}}^{n^\star} e^\star_{\mathsf{h}^\star,i}.$$

As a consequence, we have that

$$(\overline{e}^\star - \overline{e}^\star_{\mathsf{h}^\star})^{\intercal} \cdot \overline{d}^\star = (\overline{e} - \overline{e}_{(\mathsf{h}^\star \circ \mathsf{f})})^{\intercal} \cdot \overline{d} > 0.$$

Indeed, due to the above equalities and the choice of the vector $\overline{d}^\star$, it holds that

$$\overline{e}^{\star\intercal} \cdot \overline{d}^\star = \sum_{i=1}^{n^\star} e_i^\star d_i^\star = \sum_{j=1}^{n} \sum_{\substack{i=1 \\ \mathrm{g}(i)=j}}^{n^\star} e_i^\star d_i^\star$$
$$= \sum_{j=1}^{n} d_j \sum_{\substack{i=1 \\ \mathrm{g}(i)=j}}^{n^\star} e_i^\star = \sum_{j=1}^{n} e_j d_j = \overline{e}^{\intercal} \cdot \overline{d}.$$

Similarly,

$$\overline{e}^{\star\intercal}_{\mathsf{h}^\star} \cdot \overline{d}^\star = \overline{e}^{\intercal}_{(\mathsf{h}^\star \circ \mathsf{f})} \cdot \overline{d}.$$

Finally, the thesis follows by applying Theorem 4.1. □

We conclude by providing an NPTime hardness result.

Theorem 5.4. *The bag-containment problem* $q_1(\overline{x}_1) \sqsubseteq_{\mathsf{b}} q_2(\overline{x}_2)$ *of a projection-free* CQ $q_1(\overline{x}_1)$ *into a* CQ $q_2(\overline{x}_2)$ *is NPTime-hard* w.r.t. *the size of containing query.*

Proof. The proof generalizes the classic way to reduce the 3-colorability problem to the set-containment problem. Let $\mathcal{G}$ be a graph. It holds that $\mathcal{G}$ is 3-colorable *iff* there exists a homomorphism form $\mathcal{G}$ to a triangle graph $\mathcal{T}$. Now, consider the Boolean query $q_{\mathcal{G}}$ associated with $\mathcal{G}$ and the ground query $q_{\mathcal{T}} \leftarrow R(\mathsf{a},\mathsf{b}), R(\mathsf{b},\mathsf{c}), R(\mathsf{c},\mathsf{a})$ associated with $\mathcal{T}$. It holds that $\mathcal{G}$ is 3-colorable *iff* $q_{\mathcal{T}}$ is set-contained into $q_{\mathcal{G}}$. We now show that $\mathcal{G}$ is 3-colorable *iff* $q_{\mathcal{T}}$ is bag-contained into $q_{\mathcal{T}} \wedge q_{\mathcal{G}}$, as well.

(Only if). It is clear that the multiplicity of $q_{\mathcal{T}} \wedge q_{\mathcal{G}}$ is the product of the multiplicities of $q_{\mathcal{T}}$ and $q_{\mathcal{G}}$ in isolation, on any database instance, since $q_{\mathcal{T}}$ is ground. Now, suppose that $\mathcal{G}$ is 3-colorable. Then, there exists a containment mapping from $q_{\mathcal{G}}$ to $q_{\mathcal{T}}$ and, so, from $q_{\mathcal{T}} \wedge q_{\mathcal{G}}$ to $q_{\mathcal{T}}$. Therefore, it is immediate to see that $q_{\mathcal{T}}$ is bag-contained into $q_{\mathcal{T}} \wedge q_{\mathcal{G}}$, being the multiplicity of $q_{\mathcal{G}}$ greater than zero.

(If). If $q_{\mathcal{T}} \wedge q_{\mathcal{G}}$ bag-includes $q_{\mathcal{T}}$, it is obvious that $q_{\mathcal{G}}$ set-includes $q_{\mathcal{T}}$ too. Thus, $\mathcal{G}$ is 3-colorable. □

6 RELATED WORK

A very useful investigation for algebra operations on bags appeared in [17]. As discussed in the introduction, Chaudhuri and Vardi [7] posed the bag containment problem for CQs and stated, without giving a proof, that it is Π_2^{P}-hard. Since then, and to the best of our knowledge, a proof for this claim has not been provided; our NPTime hardness obviously implies the same lower bound for the general case but this bound is loose and a tighter one remains to be investigated. Chaudhuri and Vardi [7] also formulated the problem of bag containment for UCQs, proved decidability of equivalence for UCQs, as well as introduced the problem of containment for bag-set semantics, *i.e.*, when the database is a set instance but the query answers could be bags. As mentioned, undecidability of the containment problem for UCQs was proved in [20]. In [2] containment for several restrictions of CQs is studied under bag and bag-set semantics, with the problem that we close in this paper, *i.e.*, bag containment of a projection-free CQ to a CQ, surfacing as the main open problem besides the general one.

Cohen [14] studied equivalence when *combining* set and bag-set semantics within the same query and in [15] expanded to queries that combine the three semantics altogether: set, bag-set, and bag. This combination is referred to as *combined semantics* and complete characterizations of equivalence were drawn for several classes of queries that include comparisons, disjunction and atomic negation. Further building on this work, Chirkova [8–10] investigated more query classes, where equivalence, as well as minimization, is decidable or tractable under combined semantics. Recently, [11] studied equivalence for SQL queries that use a mix of bag and bag-set semantics, taking into account also possible database dependencies. More recent work that looks into bag semantics of CQs in the face of dependencies, such as views, schema mappings, or ontologies, appeared in [16, 19, 30, 31].

7 DISCUSSION

We show that the problem of checking whether a projection-free CQ is contained into a CQ under bag semantics is decidable. In particular, it can be solved in Π_2^{P} and is at least NPTime-hard. To the best of our knowledge, this has been a crucial open problem, right after the original one, which remains still open.

Our proof relies on the solution of a special case of the Diophantine inequality problem. Previous approaches that used Diophantine inequalities, notably [20] and [21], did so only for proving negative results: since the Diophantine inequality problem is undecidable, a reasonable approach is to reduce this problem to some version of bag containment in order to show undecidability of the latter. In contrast, here we focus on providing a positive result for a major class of conjunctive queries and in doing so we show decidability of an interesting restriction of the Diophantine problem. As far as we know, it is the first time that Diophantine inequalities are used as a positive tool and, in our opinion, this brings new hope for approaching the general problem. Indeed, if bag containment of CQs is eventually decidable, it must be the case that certain polynomial inequalities, and in particular those that would come out of an approach very similar to the one of Section 3, are decidable as well. Hence, a plan of attack for future works is to gradually generalize the language of the containee query and exploit the peculiarities of the structure of the associated polynomials in order to prove decidability. To do so, one can even try to reformulate the proposed technique inside the framework of functional aggregate queries [22].

Observe that the obtained mathematical results do not immediately apply to the general case, as we heavily exploit the fact that the inequalities corresponding to the studied bag containment problem compare polynomials against *monomials*. A starting point for generalization would be Lemma 4.1, which is about one variable GMPIs, and does not lift to the case of a polynomial in the right-hand side of the inequality, unless some further hypothesis is exploited that characterizes GMPIs on one unknown. In the future, we plan to further build on these results.

ACKNOWLEDGMENTS

The work by G. Konstantinidis was supported by the EU H2020 TheyBuyForYou project (780247). F. Mogavero acknowledges the support of the GNCS 2018 project "Metodi Formali per la Verica e la Sintesi di Sistemi Discreti e Ibridi".

REFERENCES

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

[2] F.N. Afrati, M. Damigos, and M. Gergatsoulis. 2010. Query Containment under Bag and Bag-Set Semantics. *IPL* 110, 10 (2010), 360–369.

[3] R.J. Brachman and H.J. Levesque. 2004. *Knowledge Representation and Reasoning.* Morgan Kaufmann.

[4] A.K. Chandra, D. Kozen, and L.J. Stockmeyer. 1981. Alternation. *JACM* 28, 1 (1981), 114–133.

[5] A.K. Chandra and P.M. Merlin. 1977. Optimal Implementation of Conjunctive Queries in Relational Data Bases.. In *STOC'77.* ACM, 77–90.

[6] A.K. Chandra and L.J. Stockmeyer. 1976. Alternation.. In *FOCS'76.* IEEECS, 98–108.

[7] S. Chaudhuri and M.Y. Vardi. 1993. Optimization of Real Conjunctive Queries.. In *PODS'93.* ACM, 59–70.

[8] R. Chirkova. 2012. Equivalence and Minimization of Conjunctive Queries under Combined Semantics.. In *ICDT'12.* OpenProceedings.org, 262–273.

[9] R. Chirkova. 2014. Combined-Semantics Equivalence and Minimization of Conjunctive Queries. *TCJ* 57, 5 (2014), 775–795.

[10] R. Chirkova. 2016. Combined-Semantics Equivalence of Conjunctive Queries: Decidability and Tractability Results. *JCSS* 82, 3 (2016), 395–465.

[11] S. Chu, A. Cheung, and D. Suciu. 2018. *Axiomatic Foundations and Algorithms for Deciding Semantic Equivalences of SQL Queries.* Technical Report. arXiv.

[12] E.F. Codd. 1970. A Relational Model of Data for Large Shared Data Banks. *CACM* 13, 6 (1970), 377–387.

[13] E.F. Codd. 1972. Relational Completeness of Data Base Sublanguages. *DS* (1972), 65–98.

[14] S. Cohen. 2006. Equivalence of Queries Combining Set and Bag-Set Semantics.. In *PODS'06.* ACM, 70–79.

[15] S. Cohen. 2009. Equivalence of Queries that are Sensitive to Multiplicities. *PVLDB* 18, 3 (2009), 765–785.

[16] M. Console, P. Guagliardo, and L. Libkin. 2017. On Querying Incomplete Information in Databases Under Bag Semantics.. In *IJCAI'17.* 993–999.

[17] S. Grumbach and T. Milo. 1996. Towards Tractable Algebras for Bags. *JCSS* 52, 3 (1996), 570–588.

[18] S. Harris, A. Seaborne, and E. Prud´hommeaux. 2013. SPARQL 1.1 query language. W3C Recommendation.

[19] A. Hernich and P.G. Kolaitis. 2017. Foundations of Information Integration under Bag Semantics.. In *LICS'17.* ACM, 1–12.

[20] Y.E. Ioannidis and R. Ramakrishnan. 1995. Containment of Conjunctive Queries: Beyond Relations as Sets. *TODS* 20, 3 (1995), 288–324.

[21] T.S. Jayram, P.G. Kolaitis, and E. Vee. 2006. The Containment Problem for Real Conjunctive Queries with Inequalities.. In *PODS'06.* ACM, 80–89.

[22] M.A. Khamis, H.Q. Ngo, and A. Rudra. 2016. FAQ: Questions Asked Frequently.. In *PODS'16.* ACM, 13–28.

[23] A. Klug. 1988. On Conjunctive Queries Containing Inequalities. *JACM* 35, 1 (1988), 146–160.

[24] P.G. Kolaitis. 2013. The Query Containment Problem: Set Semantics vs. Bag Semantics.. In *AMW'13 (CEUR-WS 1949).*

[25] G. Konstantinidis and J.L. Ambite. 2011. Scalable Query Rewriting: A Graph-Based Approach.. In *SIGMOD'11.* ACM, 97–108.

[26] S. Kopparty and B. Rossman. 2011. The Homomorphism Domination Exponent. *EJC* 32, 7 (2011), 1097–1114.

[27] A.Y. Levy, A.O. Mendelzon, Y. Sagiv, and D. Srivastava. 1995. Answering Queries Using Views.. In *PODS'95.* ACM, 95–104.

[28] Y. Matiyasevich. 1993. *Hilbert's 10th Problem.* MIT Press.

[29] G.L. Nemhauser and L.A. Wolsey. 1988. *Integer and Combinatorial Optimization.* Wiley.

[30] C. Nikolaou, E.V. Kostylev, G. Konstantinidis, M. Kaminski, B.C. Grau, and I. Horrocks. 2017. The Bag Semantics of Ontology-Based Data Access.. In *IJCAI'17.* 1224–1230.

[31] C. Nikolaou, E.V. Kostylev, G. Konstantinidis, M. Kaminski, B.C. Grau, and I. Horrocks. 2019. Foundations of Ontology-based Data Access under Bag Semantics. *AI* 274 (2019), 91–132.

[32] A. Schrijver. 1986. *Theory of Linear and Integer Programming.* Wiley.

[33] L.J. Stockmeyer. 1976. The Polynomial-Time Hierarchy. *TCS* 3, 1 (1976), 1–22.

[34] J.D. Ullman. 2000. Information Integration Using Logical Views. *TCS* 239, 2 (2000), 189–210.