Parallel-Correctness and Transferability for Conjunctive Queries under Bag Semantics

Bas Ketsman[1]
Hasselt University, Hasselt, Belgium, and transnational University of Limburg, Belgium
bas.ketsman@uhasselt.be

Frank Neven
Hasselt University, Hasselt, Belgium, and transnational University of Limburg, Belgium
frank.neven@uhasselt.be

Brecht Vandevoort[2]
Hasselt University, Hasselt, Belgium, and transnational University of Limburg, Belgium
brecht.vandevoort@uhasselt.be

Abstract

Single-round multiway join algorithms first reshuffle data over many servers and then evaluate the query at hand in a parallel and communication-free way. A key question is whether a given distribution policy for the reshuffle is adequate for computing a given query. This property is referred to as parallel-correctness. Another key problem is to detect whether the data reshuffle step can be avoided when evaluating subsequent queries. The latter problem is referred to as transfer of parallel-correctness. This paper extends the study of parallel-correctness and transfer of parallel-correctness of conjunctive queries to incorporate bag semantics. We provide semantical characterizations for both problems, obtain complexity bounds and discuss the relationship with their set semantics counterparts. Finally, we revisit both problems under a modified distribution model that takes advantage of a linear order on compute nodes and obtain tight complexity bounds.

2012 ACM Subject Classification Information systems → Query languages, Information systems → Parallel and distributed DBMSs

Keywords and phrases Conjunctive queries, distributed evaluation, bag semantics

Digital Object Identifier 10.4230/LIPIcs.ICDT.2018.18

1 Introduction

The rise of parallel data management systems like, for instance, Spark [18] and Hadoop [11], inspired a line of research on the foundations of parallel complexity of query evaluation. Several papers investigate trade-offs between the number of rounds and the amount of communication of parallel algorithms for join queries (e.g., [1–3, 6, 13, 14]). Among these, the Hypercube algorithm [3, 6, 9] is a single-round algorithm that works in two phases. The first phase is a distribution phase (where data is repartitioned or reshuffled over the servers) that is followed by a computation phase, where each server contributes to the query answer in isolation, by evaluating the query at hand over the local data without any further communication.

[1] PhD Fellow of the Research Foundation - Flanders (FWO)
[2] PhD Fellow of the Research Foundation - Flanders (FWO)

Ameloot et al. [5] introduced a framework for reasoning about generic one-round Hypercube-style algorithms for the evaluation of join queries. In this model, the distribution phase is modeled through a distribution policy specifying how the facts in the input relations are distributed among the machines. They defined two problems:

- **Parallel-Correctness:** Given a distribution policy and a query, can we be sure that the corresponding generic one-round algorithm will always compute the query result correctly, no matter the actual data?
- **Parallel-Correctness Transfer:** Given two queries $\mathcal{Q}$ and $\mathcal{Q}'$, can we infer from the fact that $\mathcal{Q}$ is computed correctly under the current distribution policy, that $\mathcal{Q}'$ is computed correctly as well?

Ameloot et al. [5] obtained tight complexity bounds for (unions of) conjunctive queries (with disequalities) for the above problems. In addition, they considered subcases that lower the complexity by either restricting the structure of queries or restricting the family of allowed distribution policies. Furthermore, it was shown (in the journal version and also in [4]) that transferability of parallel-correctness for conjunctive queries is incomparable with query containment. Geck et al. [10] consider the complexity of parallel-correctness for (unions of) conjunctive queries with negation. As a by-product it is shown that the containment problem for conjunctive queries with negation is coNEXPTIME-complete. Finally, Ketsman, Albarghouthi and Koutris [12] introduce a framework to reason about multi-round evaluation of Datalog programs and consider parallel-correctness for Datalog programs. Understanding the optimization of single-round algorithms is still important as every multi-round algorithm is a sequence of single-round steps and results from the single-round case can be transferred to or used as inspiration for studying multi-round algorithms.

Whereas the bulk of the research related to conjunctive queries focuses on set semantics, a more accurate approximation of SQL semantics is the bag semantics where multiplicities of the same tuples are taken into account. Moreover, bag semantics is particularly relevant for aggregate operators. In this paper, we therefore revisit parallel-correctness and parallel-correctness transfer under bag semantics.

As in [5], we consider conjunctive queries (CQs), allowing disequalities. Parallel-correctness under set semantics is characterized in terms of a property of minimal valuations. In brief, a CQ is parallel-correct with respect to a distribution policy if and only if for every *minimal* valuation for that query there is *at least one* compute node containing all the facts required for that valuation. Using the latter characterization, Ameloot et al. [5] obtained that testing parallel-correctness for CQs is Π_2^p-complete. In Section 3, we prove the Highlander Lemma stating that under bag semantics a CQ is parallel-correct with respect to a distribution policy if and only if for *every valuation* (not only the minimal ones) there is *exactly one* compute node containing all facts required for that valuation. Using the latter characterization, we obtain that testing for parallel-correctness under bag semantics is coNP-complete. While parallel-correctness under bag semantics implies parallel-correctness under set semantics, the converse is not true. We obtain that when CQs are strongly minimal and distribution policies are non-replicating, parallel-correctness coincides for set and bag semantics.

In a setting where multiple queries need to be evaluated, it is relevant to study whether parallel-correctness carries over from one query to another. That is, whether two queries can be evaluated after another *without* an intermediate reshuffling of the data. The latter can be relevant w.r.t. ordering of queries to improve query evaluation. For instance, in the setting of automatic data partitioning, an optimizer tries to automatically partition the base data across multiple nodes to achieve overall optimal performance for a given workload of queries (see, e.g., [15, 16]). In this setting, partitionings are thus instance dependent and not known in advance.

We say that parallel-correctness transfers from a query $\mathcal{Q}$ to a query $\mathcal{Q}'$ when $\mathcal{Q}'$ is parallel-correct under every distribution policy $\boldsymbol{P}$ under which $\mathcal{Q}$ is parallel-correct. We prove the Sandwich Lemma that provides a semantical characterization for parallel-correctness transfer under bag semantics in terms of a sandwich property for valuations. Like in the case for parallel-correctness, when comparing to set semantics, the characterization considers all valuations instead of only the minimal ones. On the other hand, as a consequence of the Highlander Lemma, the structure of queries can put additional requirements on distribution policies that are bag-parallel-correct. Therefore, our semantical characterization takes into account facts that are *implied* by a valuation w.r.t. a given query. Using the latter characterization, we obtain a decision procedure in EXPTIME for testing parallel-correctness transfer under bag semantics. In addition, we show that transferability under set and bag semantics is incomparable in general but coincides for strongly minimal conjunctive queries and non-replicating distribution policies.

The setting we have considered up to now allows every (distributed) compute node to contribute to the query result. Indeed, as is the case for the Hypercube algorithm, the result of the distributed query evaluation is the union of the results over *all* compute nodes. In this setting and under bag-semantics, the Highlander Lemma of Section 3 implies that the space of valuation for a conjunctive query should be perfectly partitioned over all compute nodes. That is, every valuation should occur in exactly one compute node. The latter can lead to situations where for particular queries the only bag-parallel-correct distribution policies are those that assign all facts to one single node. To remedy this situation, we consider the setting of ordered networks where every compute node is assigned a number and for every valuation only the node with the smallest number containing all facts required for that valuation can contribute to the query result. While both settings do not differ under set semantics, the new setting is more natural for bag semantics. We characterize parallel-correctness as well as transferability under bag semantics in this new setting and obtain tight complexity bounds.

In this paper, we make the following contributions:

1. The Highlander Lemma provides a semantical characterization of bag-parallel correctness. We obtain tight bounds for the complexity of deciding bag-parallel-correctness. We show that bag-parallel-correctness always implies set-parallel-correctness but not vice-versa and obtain that they coincide for strongly minimal queries and non-replicating distribution policies.
2. The Sandwich Lemma provides a semantical characterization of bag-parallel correctness transfer. We obtain an EXPTIME upper bound for deciding bag-parallel correctness transfer. We show that transfer of parallel-correctness under bag and set semantics is incomparable. In addition, we show that they coincide for strongly minimal queries and non-replicating distribution policies.
3. We introduce the ordered network model and again provide tight complexity bounds for parallel-correctness and transfer.

Outline

This paper is structured as follows. In Section 2, we introduce the necessary definitions. In Section 3 and Section 4, we consider parallel-correctness and parallel-correctness transfer under bag semantics. We revisit both problems under a modified distribution model that takes advantage of a linear order on compute nodes in Section 5. Finally, we conclude in Section 6.

2 Definitions

2.1 Queries and instances

We assume an infinite set **dom** of data values that are representable by strings over a fixed alphabet. A *database schema* $\mathcal{D}$ is a finite set of relation names R where every R has arity $ar(R)$. A *fact* $R(d_1, \ldots, d_k)$ is over a database schema $\mathcal{D}$ and a universe $U \subseteq \mathbf{dom}$ where $R \in \mathcal{D}$, $k = ar(R)$ and $d_1, \ldots, d_k \in U$. We use $\mathit{Facts}(\mathcal{D}, U)$ to denote the set of all facts over database schema $\mathcal{D}$ and universe $U \subseteq \mathbf{dom}$. We note that U can be infinite. We sometimes abbreviate $\mathit{Facts}(\mathcal{D}, \mathbf{dom})$ as $\mathit{Facts}(\mathcal{D})$.

An *annotated fact* $\boldsymbol{f_a}$ is a tuple $(\boldsymbol{f}, m)$ with $\boldsymbol{f}$ a fact and $m \in \mathbb{N}^+$ the multiplicity of $\boldsymbol{f}$. Here $\mathbb{N}^+$ denotes the set of strictly positive integers. A *bag of facts* F is a set of annotated facts. Every fact f may appear at most once as an annotated fact in B. That is, $(\boldsymbol{f}, m) \in B$ and $(\boldsymbol{f}', m') \in B$ implies $\boldsymbol{f} \neq \boldsymbol{f}'$. Intuitively, the multiplicity m of a fact $\boldsymbol{f}$ indicates the number of times $\boldsymbol{f}$ appears in the bag. We denote the set of facts appearing in F by $\mathit{Facts}(F)$ and the multiplicity of a fact $\boldsymbol{f}$ in the bag F by $\mathit{mul}_F(\boldsymbol{f})$. For convenience, we abuse notation and extend $\mathit{mul}_F(\boldsymbol{f})$ to arbitrary facts by setting $\mathit{mul}_F(\boldsymbol{f}) = 0$ when $\boldsymbol{f} \notin \mathit{Facts}(F)$. We next define the notion of bag union and subbag. We overload notation by using the same symbols as for set union and subset. It should always be clear from the context whether we refer to bags or to sets. For two bags of facts F and G, the *bag union*, denoted $F \cup G$, is defined as $\mathit{Facts}(F) \cup \mathit{Facts}(G)$ and $\mathit{mul}_H(\boldsymbol{f}) = \mathit{mul}_F(\boldsymbol{f}) + \mathit{mul}_G(\boldsymbol{f})$ for each fact $\boldsymbol{f} \in \mathit{Facts}(H)$. Furthermore, F is a *subbag* of G, denoted $F \subseteq G$, if $\mathit{mul}_F(\boldsymbol{f}) \leq \mathit{mul}_G(\boldsymbol{f})$ for each fact $\boldsymbol{f} \in \mathit{Facts}(F)$. By $|F|$, we denote the number of facts in F, that is, $\sum_{\boldsymbol{f} \in \mathit{Facts}(F)} \mathit{mul}_F(\boldsymbol{f})$.

A *database instance* I, instance for short, over a database schema $\mathcal{D}$ is a bag of facts, with $\mathit{Facts}(I) \subseteq \mathit{Facts}(\mathcal{D})$. We use $\mathit{adom}(I)$ to denote the set of data values occurring in I.

A *query* $\mathcal{Q}$ *over input schema* $\mathcal{D}_1$ *and output schema* $\mathcal{D}_2$ is a generic mapping from instances over $\mathcal{D}_1$ to instances over $\mathcal{D}_2$. A query $\mathcal{Q}$ is *monotone* if $\mathcal{Q}(I') \subseteq \mathcal{Q}(I)$ for every pair of instances I and I' with $I' \subseteq I$.

2.2 Conjunctive queries

Assume an infinite set of variables **var**, disjoint from **dom**. An *atom* over a database schema $\mathcal{D}$ is of the form $R(\mathbf{x})$, with $R \in \mathcal{D}$ and $\mathbf{x} = (x_1, \ldots, x_k)$ a tuple of variables in **var** with $k = ar(R)$.

A *conjunctive query* $\mathcal{Q}$ over input schema $\mathcal{D}$ is an expression of the form

$$T(\mathbf{x}) \leftarrow R_1(\mathbf{y_1}), \ldots, R_m(\mathbf{y_m}), \beta_1, \ldots, \beta_p$$

where every $R_i(\mathbf{y_i})$ is an atom over $\mathcal{D}$, $T(\mathbf{x})$ is an atom, called the *head atom*, with $T \notin \mathcal{D}$, and every β_i is a disequality of the form $z \neq z'$ (with z a variable different from z'). Every variable $x \in \mathbf{x}$ needs to appear in at least one $\mathbf{y_i}$. We require that every variable occurring in a disequality occurs in at least one $\mathbf{y_i}$. Furthermore, we refer to $T(\mathbf{x})$ as $\mathit{head}_{\mathcal{Q}}$, to the set $\{R_1(\mathbf{y_1}), \ldots, R_m(\mathbf{y_m})\}$ as $\mathit{body}_{\mathcal{Q}}$ and to the set of all variables occurring in $\mathcal{Q}$ as $\mathit{vars}(\mathcal{Q})$.

We denote by $\mathbf{CQ}^{\neq}$ the set of all conjunctive queries (allowing disequalities) and by **CQ** the set of conjunctive queries without disequalities. A conjunctive query with disequalities is *without self-joins* if all of its atoms have distinct relation names. A conjunctive query with disequalities $\mathcal{Q}$ is *full* if every variable occurring in $\mathcal{Q}$ appears in the head atom.

A *valuation* for a conjunctive query $\mathcal{Q} \in \mathbf{CQ}^{\neq}$ is a total function $V : \mathit{vars}(\mathcal{Q}) \to \mathbf{dom}$ that is consistent with the disequalities in $\mathcal{Q}$. More specifically: for every $z \neq z'$ in $\mathcal{Q}$ it holds that $V(z) \neq V(z')$. Valuations naturally extend to atoms and sets of atoms. We refer to $V(body_{\mathcal{Q}})$ as the set of facts *required* by V.

A valuation V *satisfies* a conjunctive query $\mathcal{Q} \in \mathbf{CQ}^{\neq}$ on instance I if $V(body_{\mathcal{Q}}) \subseteq Facts(I)$. In that case, V *derives the annotated fact* $\boldsymbol{f_a} = (V(head_{\mathcal{Q}}), m)$, with

$$m = \prod_{\boldsymbol{f} \in V(body_{\mathcal{Q}})} mul_I(\boldsymbol{f}).$$

For convenience, we also say that V *derives* the fact $\boldsymbol{f} = V(head_{\mathcal{Q}})$ if V satisfies $\mathcal{Q}$ on I. The result of V on an instance I, denoted $[\mathcal{Q}, V](I)$, is the bag of annotated facts derived by V on instance I. This bag is empty when V does not satisfy $\mathcal{Q}$ on I. When V does satisfy $\mathcal{Q}$ on I, the set $Facts([\mathcal{Q}, V](I))$ is always a singleton. The *result* $\mathcal{Q}(I)$ of a conjunctive query $\mathcal{Q} \in \mathbf{CQ}^{\neq}$ on I is defined as the bag union over all results of satisfying valuations for $\mathcal{Q}$ on I:

$$\mathcal{Q}(I) = \bigcup_{V \in \mathcal{V}} [\mathcal{Q}, V](I)$$

with $\mathcal{V}$ the set containing all valuations that satisfy $\mathcal{Q}$ on I.

2.3 Networks, data distribution and policies

A *network* $\mathcal{N}$ is a nonempty finite set of values from **dom**, called *nodes*.

A distribution policy specifies how a database, possibly already distributed, is reshuffled by determining which fact is sent to which server. Formally, a *distribution policy* $\boldsymbol{P} = (U, rfacts_{\boldsymbol{P}})$ for a database schema $\mathcal{D}$ and a network $\mathcal{N}$ consists of a universe U and a total function $rfacts_{\boldsymbol{P}} : \mathcal{N} \to 2^{Facts(\mathcal{D},U)}$ mapping each node $\kappa \in \mathcal{N}$ onto a set of facts from $Facts(\mathcal{D}, U)$. A node $\kappa \in \mathcal{N}$ is *responsible* for a fact $\boldsymbol{f} \in Facts(\mathcal{D}, U)$ under $\boldsymbol{P}$ if $\boldsymbol{f} \in rfacts_{\boldsymbol{P}}(\kappa)$. For an instance I, the function $loc\text{-}inst_{\boldsymbol{P},I}$ maps each node $\kappa \in \mathcal{N}$ to the bag of facts it is responsible for. More formally, $(\boldsymbol{f}, m) \in loc\text{-}inst_{\boldsymbol{P},I}(\kappa)$ iff $(\boldsymbol{f}, m) \in I$ and $\boldsymbol{f} \in rfacts_{\boldsymbol{P}}(\kappa)$. We refer to I as the *global instance* and to $loc\text{-}inst_{\boldsymbol{P},I}(\kappa)$ as the *local instance at node* κ.

As distribution policies are defined on facts, either all copies of a certain fact are sent to a specific server or none are. The latter happens for instance when using hash functions to define distribution policies as is the case for instance for Hypercube [3, 6, 9].

Next, we define the one-round distributed evaluation induced by $\boldsymbol{P}$. Query $\mathcal{Q}$ is evaluated at each node κ separately, after which the bag union of all results is taken:

$$[\mathcal{Q}, \boldsymbol{P}](I) = \bigcup_{\kappa \in \mathcal{N}} \mathcal{Q}(loc\text{-}inst_{\boldsymbol{P},I}(\kappa)).$$

2.4 Classes of distribution policies

To reason about the complexity of problems involving distribution policies (which are just defined as functions), we need to consider a representation mechanism for these policies. For this, we first discuss the classes $\mathcal{P}_{fin}$ and $\mathfrak{P}_{nondet}$ as introduced by Ameloot et al. [5] and then describe the class $\mathfrak{P}_{det}$.

The class $\mathcal{P}_{fin}$ is defined over distribution policies with a finite universe. Intuitively, $\mathcal{P}_{fin}$ allows to express all distribution policies over a finite universe, but uses the most naive and exhaustive representation mechanism: explicit enumeration. Formally, a policy $\boldsymbol{P} = (U, rfacts_{\boldsymbol{P}})$ belongs to $\mathcal{P}_{fin}$ if U is a finite set. Such policies are represented by an explicit enumeration of the data values in U and an explicit enumeration of all pairs $(\kappa, \boldsymbol{f})$ where $\boldsymbol{f} \in rfacts_{\boldsymbol{P}}(\kappa)$.

A more general way to describe classes of distribution policies by an arbitrarily succinct representation is by means of a "test algorithm" that allows to decide $\boldsymbol{f} \in \mathit{rfacts}_{\boldsymbol{P}}(\kappa)$ with time bound ℓ^k, where ℓ is the length of the input and k a constant. We call this class $\mathfrak{P}_{nondet}$. More precisely, a policy $\boldsymbol{P} = (U, \mathit{rfacts}_{\boldsymbol{P}})$ over network $\mathcal{N}$ is in $\mathcal{P}^k_{nondet}$ if it is specified by a pair $(n, \mathcal{A}_P)$, with n a natural number in unary representation and $\mathcal{A}_P$ a non-deterministic algorithm. The value n is used to give an upper bound to the length of data values in universe U and on the names of nodes in $\mathcal{N}$. More specifically, the universe U consists of all data values representable by a string of length at most n and the network $\mathcal{N}$ consists of all nodes representable by strings of length at most n. A fact f is in $\mathit{rfacts}_{\boldsymbol{P}}(\kappa)$ for a given node κ if $\mathcal{A}_P$ has an accepting run of at most $|(\kappa, \boldsymbol{f})|^k$ steps on input $(\kappa, \boldsymbol{f})$. We define $\mathfrak{P}_{nondet}$ as the set $\{\mathcal{P}^k_{nondet} \mid k \geq 2\}$. We remark that each policy in $\mathcal{P}_{\mathit{fin}}$ can thus be described in $\mathcal{P}^2_{nondet}$.

The complexity of deciding set-parallel-correctness is so high that complexity bounds are retained even when considering policies in $\mathcal{P}^k_{nondet}$. For bag-parallel-correctness this is not the case and considering policies from $\mathcal{P}^k_{nondet}$ artificially increases the complexity of the decision problem. Therefore, for bag-parallel-correctness, we use the class $\mathcal{P}^k_{det}$, which is defined next. A policy $\boldsymbol{P} = (U, \mathit{rfacts}_{\boldsymbol{P}})$ is in $\mathcal{P}^k_{det}$ if it can be specified by a tuple $(\mathcal{N}, n, \mathcal{A}_P)$ where $\mathcal{N}$ is an explicit enumeration of the nodes in the network, n is a natural number in unary representation and $\mathcal{A}_P$ is a *deterministic* algorithm. The universe U of $\boldsymbol{P}$ is the set of values representable by strings of length at most n. Given a fact $\boldsymbol{f}$ and node κ, algorithm $\mathcal{A}_P$ decides in at most $|(\kappa, \boldsymbol{f})|^k$ steps whether $\boldsymbol{f} \in \mathit{rfacts}_{\boldsymbol{P}}(\kappa)$. We define $\mathfrak{P}_{det}$ as the set of policies $\{\mathcal{P}^k_{det} \mid k \geq 2\}$.

Since each distribution policy implicitly induces a network and each query implicitly defines a database schema, we often omit the explicit notation for networks and schemas.

3 Parallel-correctness

Intuitively, the notion of parallel-correctness relates to whether the distributed execution of a query with relation to a specific distribution policy produces the correct result. That is, whether the distributed execution produces the same result as when the query was evaluated on the global instance.

3.1 Definition and results for set-parallel-correctness

We distinguish between parallel-correctness under the set and under the bag semantics. The former was introduced in [5] and we refer to it as set-parallel-correctness. We next generalize the notion to bag semantics and call it bag-parallel-correctness. Recall that $\mathit{Facts}(F)$ denotes the set of facts occurring in the bag F.

▶ **Definition 3.1.** Let $\mathcal{Q}$ be a query and $\boldsymbol{P}$ a distribution policy. Then,

- $\mathcal{Q}$ is *bag-parallel-correct on instance I under $\boldsymbol{P}$* if $\mathcal{Q}(I) = [\mathcal{Q}, \boldsymbol{P}](I)$;
- $\mathcal{Q}$ is *set-parallel-correct on instance I under $\boldsymbol{P}$* if $\mathit{Facts}(\mathcal{Q}(I)) = \mathit{Facts}([\mathcal{Q}, \boldsymbol{P}](I))$; and,
- $\mathcal{Q}$ is *bag-parallel-correct (resp., set-) under $\boldsymbol{P}$* if $\mathcal{Q}$ is bag-parallel-correct (resp., set-) on all instances I under $\boldsymbol{P}$.

We now formally define the decision problems related to parallel-correctness. In the following, $\mathcal{C}$ denotes a query class, $\mathcal{P}$ denotes a class of distribution policies, and $x \in \{\text{set}, \text{bag}\}$. Then, define the following problem definitions:

	$\mathbf{PCI}^x(\mathcal{C}, \mathcal{P}, \mathbf{I})$
Input:	Query $\mathcal{Q} \in \mathcal{C}$, distribution policy $\boldsymbol{P} \in \mathcal{P}$, instance I
Question:	Is $\mathcal{Q}$ x-parallel-correct on I under $\boldsymbol{P}$?

	$\mathbf{PC}^x(\mathcal{C}, \mathcal{P})$
Input:	Query $\mathcal{Q} \in \mathcal{C}$, distribution policy $\boldsymbol{P} \in \mathcal{P}$
Question:	Is $\mathcal{Q}$ x-parallel-correct under $\boldsymbol{P}$?

We recall the following result by Ameloot et al. [5]:

▶ **Theorem 3.2** ([5]). *Problems $\mathbf{PCI}^{set}(\mathcal{C}, \mathcal{P})$ and $\mathbf{PC}^{set}(\mathcal{C}, \mathcal{P})$ are Π_2^p-complete for every query class $\mathcal{C} \in \{\mathbf{CQ}, \mathbf{CQ}^{\neq}\}$ and for every policy class $\mathcal{P} \in \{\mathcal{P}_{fin}\} \cup \mathfrak{P}_{nondet}$.*

The upper bounds given by the above theorem follow rather directly from the semantical characterization given in the next lemma. To this end, we need the notion of minimal valuations. For $\mathcal{Q}$ in $\mathbf{CQ}^{\neq}$, a valuation V is *minimal* if there is *no valuation* V' for $\mathcal{Q}$ that derives the same head fact with a strict subset of body facts, that is, such that $V'(body_{\mathcal{Q}}) \subsetneq V(body_{\mathcal{Q}})$ and $V'(head_{\mathcal{Q}}) = V(head_{\mathcal{Q}})$. Recall from the definitions that $V(body_{\mathcal{Q}})$ always refers to a set of facts, regardless of the considered semantics.

▶ **Lemma 3.3** ([5]). *Let $\mathcal{Q}$ be in $\mathbf{CQ}^{\neq}$. Then $\mathcal{Q}$ is set-parallel-correct under distribution policy $\boldsymbol{P} = (U, rfacts_{\boldsymbol{P}})$ if and only if for every minimal valuation V for $\mathcal{Q}$ over U, there is a node $\kappa \in \mathcal{N}$ such that $V(body_{\mathcal{Q}}) \subseteq rfacts_{\boldsymbol{P}}(\kappa)$.*

3.2 Bag-parallel-correctness

We now discuss the problem of deciding bag-parallel-correctness. To start, we obtain a property that characterizes bag-parallel-correctness in direct analogy to Lemma 3.3. The characterization for bag-parallel-correctness is again related to valuations but is more strict than the condition of Lemma 3.3 in two different ways. First, the condition should now hold for *all* valuations not just the minimal ones. Second, the condition requires that, for each valuation, *there can be only one* node harboring all the required facts for that valuation.

To prove the next lemma, we introduce the notion of support. For $\mathcal{Q} \in \mathbf{CQ}^{\neq}$ and distribution policy $\boldsymbol{P}$, we say that node κ *supports* valuation V for $\mathcal{Q}$, if $V(body_{\mathcal{Q}}) \subseteq rfacts_{\boldsymbol{P}}(\kappa)$. By $Sup_{\boldsymbol{P}}(\mathcal{Q}, V)$, we denote the set of all nodes that support V under $\boldsymbol{P}$.

▶ **Lemma 3.4** (Highlander Lemma[3]). *For $\mathcal{Q} \in \mathbf{CQ}^{\neq}$ and a distribution policy $\boldsymbol{P} = (U, rfacts_{\boldsymbol{P}})$ over $\mathcal{N}$, $\mathcal{Q}$ is bag-parallel-correct under $\boldsymbol{P}$ if and only if $|Sup_{\boldsymbol{P}}(\mathcal{Q}, V)| = 1$, for every valuation V for $\mathcal{Q}$.*

Proof sketch. *(If).* Since every valuation V for $\mathcal{Q}$ is supported by exactly one node, bag-parallel-correctness follows trivially.

(Only-if). Let $\mathcal{Q}$ be bag-parallel-correct for $\boldsymbol{P}$. We claim that for all valuations V for $\mathcal{Q}$, $|Sup_{\boldsymbol{P}}(\mathcal{Q}, V)| \leq 1$. Indeed, if there is a valuation V with $|Sup_{\boldsymbol{P}}(\mathcal{Q}, V)| > 1$ and $\boldsymbol{f} = V(head_{\mathcal{Q}})$, it follows that the multiplicity of $\boldsymbol{f}$ in $[\mathcal{Q}, \boldsymbol{P}](I)$ with $Facts(I) = V(body_{\mathcal{Q}})$ is too high.

It remains to argue that there cannot be a valuation V for $\mathcal{Q}$ with $|Sup_{\boldsymbol{P}}(\mathcal{Q}, V)| = 0$. Assume towards a contradiction that such a valuation V exists, and let $\boldsymbol{f} = V(head_{\mathcal{Q}})$. For

[3] “There can be only one.” `https://en.wikipedia.org/wiki/Highlander_(film)`

an arbitrary instance I with $\mathit{Facts}(I) = V(\mathit{body}_{\mathcal{Q}})$, the resulting multiplicity of $\boldsymbol{f}$ in $[\mathcal{Q}, \boldsymbol{P}](I)$ is too low, unless there is a valuation W for $\mathcal{Q}$ with $\boldsymbol{f} = W(\mathit{head}_{\mathcal{Q}})$ and $|\mathit{Sup}_{\boldsymbol{P}}(\mathcal{Q}, W)| > 1$. But this contradicts our earlier claim. ◀

We next obtain the complexity of deciding bag-parallel-correctness. The upper bound follows rather directly from Lemma 3.4. The lower bound is a reduction from the complement of 3-SAT.

▶ **Theorem 3.5.** $\mathbf{PC}^{bag}(\mathcal{C}, \mathcal{P})$ *is* coNP*-complete for every query class* $\mathcal{C} \in \{\mathbf{CQ}, \mathbf{CQ}^{\neq}\}$ *and every policy class* $\mathcal{P} \in \{\mathcal{P}_{\mathit{fin}}\} \cup \mathfrak{P}_{det}$*, even over networks with only two nodes.*

3.3 Relationship between set- and bag-parallel-correctness

We next address the relationship between set- and bag-parallel-correctness. The implication in the following proposition follows immediately from Lemma 3.3 and Lemma 3.4. A counterexample for the converse is given in Example 3.7.

▶ **Proposition 3.6.** *Bag-parallel-correctness implies set-parallel-correctness for queries in* $\mathbf{CQ}^{\neq}$*, but not vice-versa.*

▶ **Example 3.7.** For an example showing that the reverse direction of Proposition 3.6 does not hold, consider query $\mathcal{Q}$: $T(x) \leftarrow R(x)$. Let $\boldsymbol{P} = (U, \mathit{rfacts}_{\boldsymbol{P}})$ be a distribution policy over network $\mathcal{N} = \{\kappa_1, \kappa_2\}$, with $\mathit{rfacts}_{\boldsymbol{P}}(\kappa_1) = \mathit{rfacts}_{\boldsymbol{P}}(\kappa_2) = \{R(a), R(b)\}$, and $U = \{a, b\}$.

We observe that $\mathcal{Q}$ has only two valuations under U which in addition are minimal: $V_a = \{x \mapsto a\}$ and $V_b = \{x \mapsto b\}$. Since $\mathit{Sup}_{\boldsymbol{P}}(\mathcal{Q}, V_1) = \mathit{Sup}_{\boldsymbol{P}}(\mathcal{Q}, V_2) = \{\kappa_1, \kappa_2\}$ it follows immediately from Lemma 3.3 and Lemma 3.4 that $\mathcal{Q}$ is set-parallel-correct, but not bag-parallel-correct, under $\boldsymbol{P}$. ◀

Interestingly, we can identify a class of $\mathbf{CQ}^{\neq}$-queries and a class of distribution policies for which the notions of set- and bag-parallel-correctness coincide. First, we introduce the necessary definitions.

A query in $\mathbf{CQ}^{\neq}$ is *strongly minimal* if all its valuations are minimal. We consider the family of non-replicating distribution policies that do not replicate any fact onto multiple nodes. More formally, a distribution policy $\boldsymbol{P} = (U, \mathit{rfacts}_{\boldsymbol{P}})$ over a network $\mathcal{N}$ is *non-replicating* if and only if $\mathit{rfacts}_{\boldsymbol{P}}(\kappa_1) \cap \mathit{rfacts}_{\boldsymbol{P}}(\kappa_2) = \emptyset$ for every pair of nodes $\kappa_1, \kappa_2 \in \mathcal{N}$ with $\kappa_1 \neq \kappa_2$.

▶ **Theorem 3.8.** *For a strongly minimal query* $\mathcal{Q}$ *in* $\mathbf{CQ}^{\neq}$ *and a non-replicating distribution policy* $\boldsymbol{P}$*,* $\mathcal{Q}$ *is bag-parallel-correct under* $\boldsymbol{P}$ *iff* $\mathcal{Q}$ *is set-parallel-correct under* $\boldsymbol{P}$*.*

Proof. It follows from Proposition 3.6 that bag-parallel-correctness of $\mathcal{Q}$ under $\boldsymbol{P}$ implies set-parallel-correctness. We show the reverse direction through Lemma 3.4. For this, let V be an arbitrary valuation for $\mathcal{Q}$. Since $\mathcal{Q}$ is set-parallel-correct under $\boldsymbol{P}$, and V is minimal (due to strong minimality of $\mathcal{Q}$), it follows from Lemma 3.3 that $|\mathit{Sup}_{\boldsymbol{P}}(\mathcal{Q}, V)| \geq 1$. Since $\boldsymbol{P}$ is non-replicating, the latter implies $|\mathit{Sup}_{\boldsymbol{P}}(\mathcal{Q}, V)| = 1$. ◀

Notice that in the constructed counterexample from Example 3.7, the query $\mathcal{Q}$ is strongly minimal, but $\boldsymbol{P}$ is replicating. In the following example we show that, for Theorem 3.8, the condition that $\mathcal{Q}$ is strongly minimal can not be dropped.

▶ **Example 3.9.** Consider query $\mathcal{Q}$: $T(x) \leftarrow R(x), R(y)$, and network $\mathcal{N} = \{\kappa_1, \kappa_2\}$. Let $\boldsymbol{P} = (U, \mathit{rfacts}_{\boldsymbol{P}})$ be a distribution policy over $U = \{a, b\}$ and $\mathcal{N}$, with $\mathit{rfacts}_{\boldsymbol{P}}(\kappa_1) = \{R(a)\}$ and $\mathit{rfacts}_{\boldsymbol{P}}(\kappa_2) = \{R(b)\}$. Notice that $\boldsymbol{P}$ is non-replicating.

We observe that $\boldsymbol{P}$ is set-parallel-correct for $\mathcal{Q}$. Indeed, there are only two minimal valuations for $\mathcal{Q}$ over U: $V_a = \{x \mapsto a, y \mapsto a\}$ and $V_b = \{x \mapsto b, y \mapsto b\}$. Furthermore, V_a is supported by κ_1 while V_b is supported by κ_2. The result then follows from Lemma 3.3.

For non-minimal valuation $V = \{x \mapsto a, y \mapsto b\}$, we observe that $|Sup_{\boldsymbol{P}}(\mathcal{Q}, V)| = \emptyset$. Thus $\boldsymbol{P}$ cannot be bag-parallel-correct for $\mathcal{Q}$ (due to Lemma 3.4). ◀

4 Transferability

Parallel-correctness *transfers* from a query $\mathcal{Q}$ to a query $\mathcal{Q}'$ when $\mathcal{Q}'$ is parallel-correct under every distribution policy $\boldsymbol{P}$ under which $\mathcal{Q}$ is parallel-correct. This means in particular that query $\mathcal{Q}'$ can *always* be evaluated after query $\mathcal{Q}$ *without* an intermediate, possibly expensive, reshuffling of the data. The present section studies parallel-correctness transfer under bag semantics.

4.1 Definition and results for transferability under set semantics

The notion of parallel-correctness transfer was introduced by Ameloot et al. [5]. We next distinguish between transferability under set and bag semantics.

▶ **Definition 4.1.** For two queries $\mathcal{Q}$ and $\mathcal{Q}'$ over the same input schema, *bag-parallel-correctness transfers from $\mathcal{Q}$ to $\mathcal{Q}'$* if $\mathcal{Q}'$ is bag-parallel-correct under every distribution policy for which $\mathcal{Q}$ is bag-parallel-correct. In this case, we write $\mathcal{Q} \xrightarrow{\text{bag}} \mathcal{Q}'$. Set-parallel-correctness transferability is defined similarly and denoted by $\mathcal{Q} \xrightarrow{\text{set}} \mathcal{Q}'$.

▶ **Lemma 4.2** ([5]). *For queries $\mathcal{Q}, \mathcal{Q}' \in \mathbf{CQ}^{\neq}$, set-parallel-correctness transfers from $\mathcal{Q}$ to $\mathcal{Q}'$ if for each minimal valuation V' for $\mathcal{Q}'$ there is a minimal valuation V for $\mathcal{Q}$ where $V'(body_{\mathcal{Q}'}) \subseteq V(body_{\mathcal{Q}})$ and $adom(V'(body_{\mathcal{Q}'})) = adom(V(body_{\mathcal{Q}}))$.*

4.2 Transferability under bag semantics

The following example highlights how, depending on the structure of the query, different valuations must be supported by the *same* compute node for distribution policies under which the query is bag-parallel-correct. In particular, the example shows that the assignment of a fact to a particular node can *imply* that other facts should be assigned to that same node as well.

▶ **Example 4.3.** Consider the query $\mathcal{Q} : H(x) \leftarrow R(x, y), R(x, z)$. Let $\boldsymbol{P}$ be a distribution policy under which $\mathcal{Q}$ is bag-parallel-correct. Assume $R(a, a) \in rfacts_{\boldsymbol{P}}(\kappa)$ for some node κ. Then, by Lemma 3.4, every fact of the form $R(a, c)$ for any c should belong to $rfacts_{\boldsymbol{P}}(\kappa)$ as well. Furthermore, denoting the valuation $\{x \mapsto a, y \mapsto b, z \mapsto c\}$ by $W_{a,b,c}$, the following set of valuations $\{W_{a,b,c} \mid b, c \in U\}$ for a fixed a have to be supported by the same node. ◀

We formally define the set of facts that are implied by a valuation w.r.t. a given query.

▶ **Definition 4.4.** Let V be a valuation for $\mathcal{Q} \in \mathbf{CQ}^{\neq}$. A fact $\boldsymbol{f}$ is *implied* by V w.r.t. $\mathcal{Q}$ if for every distribution policy $\boldsymbol{P} = (U, rfacts_{\boldsymbol{P}})$, with $adom(V(body_{\mathcal{Q}})) \subseteq U$ under which $\mathcal{Q}$ is bag-parallel-correct, and for every node κ in the network of $\boldsymbol{P}$: $V(body_{\mathcal{Q}}) \subseteq rfacts_{\boldsymbol{P}}(\kappa)$ implies $\boldsymbol{f} \in rfacts_{\boldsymbol{P}}(\kappa)$. We denote the set of facts implied by V w.r.t. $\mathcal{Q}$ by $ImpFacts(V, \mathcal{Q})$.

Notice that $ImpFacts(V, \mathcal{Q})$ is well-defined as there is always a distribution policy under which Q is bag-parallel-correct: namely, the policy which is defined over a single-node network and maps all facts to a single node. Furthermore, $\mathit{ImpFacts}(V, \mathcal{Q}) \subseteq \mathit{rfacts}_{\boldsymbol{P}}(\kappa)$ whenever $V(\mathit{body}_{\mathcal{Q}}) \subseteq \mathit{rfacts}_{\boldsymbol{P}}(\kappa)$ for every distribution policy $\boldsymbol{P}$ under which $\mathcal{Q}$ is bag-parallel-correct.

We are now ready to characterize bag-parallel-correctness transfer. The lemma plays a role similar to the Highlander Lemma and requires that every valuation for the second query is sandwiched between a valuation for the first query and the implied facts.

▶ **Lemma 4.5** (Sandwich lemma). *Bag-parallel-correctness transfers from $\mathcal{Q}$ to $\mathcal{Q}'$ if and only if for each valuation V' for $\mathcal{Q}'$ there is a valuation V for $\mathcal{Q}$ such that $V(\mathit{body}_{\mathcal{Q}}) \subseteq V'(\mathit{body}_{\mathcal{Q}'}) \subseteq \mathit{ImpFacts}(V, \mathcal{Q})$.*

Proof. *(If).* Let $\boldsymbol{P} = (U, \mathit{rfacts}_{\boldsymbol{P}})$ be an arbitrary distribution policy such that $\mathcal{Q}$ is bag-parallel-correct under $\boldsymbol{P}$. Let V' be an arbitrary valuation for $\mathcal{Q}'$ over U. We argue that $|\mathit{Sup}_{\boldsymbol{P}}(\mathcal{Q}', V')| = 1$ which by Lemma 3.4 implies that $\mathcal{Q}'$ is bag-parallel-correct under $\boldsymbol{P}$ as well. By assumption there is a valuation V for $\mathcal{Q}$ over U such that $V(\mathit{body}_{\mathcal{Q}}) \subseteq V'(\mathit{body}_{\mathcal{Q}'}) \subseteq \mathit{ImpFacts}(V, \mathcal{Q})$. Then, by Lemma 3.4, $\mathit{Sup}_{\boldsymbol{P}}(\mathcal{Q}, V) = \{\kappa\}$ for some node κ and $\mathit{ImpFacts}(V, \mathcal{Q}) \subseteq \mathit{rfacts}_{\boldsymbol{P}}(\kappa)$. Therefore, $V'(\mathit{body}_{\mathcal{Q}'}) \subseteq \mathit{rfacts}_{\boldsymbol{P}}(\kappa)$. So, $|\mathit{Sup}_{\boldsymbol{P}}(\mathcal{Q}', V')| \geq 1$. However, as $V(\mathit{body}_{\mathcal{Q}}) \subseteq \mathit{rfacts}_{\boldsymbol{P}}(\kappa)$ and $\mathit{Sup}_{\boldsymbol{P}}(\mathcal{Q}, V) = \{\kappa\}$, $|\mathit{Sup}_{\boldsymbol{P}}(\mathcal{Q}', V')| = 1$.

(Only-If). The proof is by contraposition. In particular, we show that bag-parallel-correctness does not transfer from $\mathcal{Q}$ to $\mathcal{Q}'$ if the condition of the lemma fails for some valuation V' for $\mathcal{Q}'$. We distinguish two cases: the case when no valuation V for $\mathcal{Q}$ exists with $V(\mathit{body}_{\mathcal{Q}}) \subseteq V'(\mathit{body}_{\mathcal{Q}'})$, and the case when for each valuation V, with $V(\mathit{body}_{\mathcal{Q}}) \subseteq V'(\mathit{body}_{\mathcal{Q}'})$, we have that $V'(\mathit{body}_{\mathcal{Q}'}) \not\subseteq \mathit{ImpFacts}(V, \mathcal{Q})$.

Case 1: there is no valuation V with $V(\mathit{body}_{\mathcal{Q}}) \subseteq V'(\mathit{body}_{\mathcal{Q}'})$. We construct the policy $\boldsymbol{P}$ over a two-node network $\{\kappa_1, \kappa_2\}$ and universe U consisting of all domain values used by V', with $\mathit{rfacts}_{\boldsymbol{P}}(\kappa_1) = \mathit{Facts}(\mathcal{D}, U)$ and $\mathit{rfacts}_{\boldsymbol{P}}(\kappa_2) = V'(\mathit{body}_{\mathcal{Q}'})$. Then, $\mathit{Sup}_{\boldsymbol{P}}(Q', V') = \{\kappa_1, \kappa_2\}$ and Lemma 3.4 implies that $\boldsymbol{P}$ is not bag-parallel-correct for $\mathcal{Q}'$. In contrast, every valuation for $\mathcal{Q}$ is supported only on node κ_1 (as none of them are included in $V'(\mathit{body}_{\mathcal{Q}'})$) which implies that $\boldsymbol{P}$ is bag-parallel-correct for $\mathcal{Q}$. We conclude that bag-parallel-correctness does not transfer from $\mathcal{Q}$ to $\mathcal{Q}'$.

Case 2: for each valuation V, $V(\mathit{body}_{\mathcal{Q}}) \subseteq V'(\mathit{body}_{\mathcal{Q}'})$ implies $V'(\mathit{body}_{\mathcal{Q}'}) \not\subseteq \mathit{ImpFacts}(V, \mathcal{Q})$. From the previous case, we can assume the existence of a valuation V with $V(\mathit{body}_{\mathcal{Q}}) \subseteq V'(\mathit{body}_{\mathcal{Q}'})$. Then, by definition of $\mathit{ImpFacts}(V, \mathcal{Q})$, $V'(\mathit{body}_{\mathcal{Q}'}) \not\subseteq \mathit{ImpFacts}(V, \mathcal{Q})$ implies that there must be a policy $\boldsymbol{P}$ (over some network $\mathcal{N}$) such that $\mathcal{Q}$ is bag-parallel-correct under $\boldsymbol{P}$ and $\boldsymbol{P}$ has a node κ with $V(\mathit{body}_{\mathcal{Q}}) \subseteq \mathit{rfacts}_{\boldsymbol{P}}(\kappa)$ and $V'(\mathit{body}_{\mathcal{Q}'}) \not\subseteq \mathit{rfacts}_{\boldsymbol{P}}(\kappa)$. From Lemma 3.4, it follows that for all other nodes κ', that is $\kappa' \in \mathcal{N} \setminus \{\kappa\}$, $V(\mathit{body}_{\mathcal{Q}}) \not\subseteq \mathit{rfacts}_{\boldsymbol{P}}(\kappa')$, and thus $V'(\mathit{body}_{\mathcal{Q}'}) \not\subseteq \mathit{rfacts}_{\boldsymbol{P}}(\kappa')$. Hence, $\boldsymbol{P}$ is not bag-parallel-correct for $\mathcal{Q}'$ and, consequently, bag-parallel-correctness does not transfer from $\mathcal{Q}$ to $\mathcal{Q}'$. ◀

Notice that the inclusion between $V(\mathit{body}_{\mathcal{Q}})$ and $V'(\mathit{body}_{\mathcal{Q}'})$ in Lemma 4.5 is in the opposite direction as in Lemma 4.2, since the inclusion now asserts that V' is supported by at most one node instead of at least one.

We formally define the respective decision problems for $x \in \{\text{set}, \text{bag}\}$. By $\mathcal{C}$ and $\mathcal{C}'$ we denote query classes.

	PC-Trans$^x(\mathcal{C}, \mathcal{C}')$
Input:	Query $\mathcal{Q} \in \mathcal{C}$, query $\mathcal{Q}' \in \mathcal{C}'$
Question:	Does x-parallel-correctness transfer from $\mathcal{Q}$ to $\mathcal{Q}'$?

Algorithm 1 MAX-PROOF-FOREST($\mathcal{Q}, U$).

Let $\mathcal{I}$ be the set of single-node IF-proof-trees, one for each set $V(body_{\mathcal{Q}})$, where V is a valuation for $\mathcal{Q}$ over U.
while Distinct $\mathbf{T}_1, \mathbf{T}_2 \in \mathcal{I}$ and V for $\mathcal{Q}$ over U exist, with $V(body_{\mathcal{Q}}) \subseteq Inst_{\mathbf{T}_1}(n_1) \cap Inst_{\mathbf{T}_2}(n_2)$, with n_1, n_2 the roots of $\mathbf{T}_1, \mathbf{T}_2$ respectively **do**
  Remove $\mathbf{T}_1$ and $\mathbf{T}_2$ from $\mathcal{I}$
  Insert new node n with children $\mathbf{T}_1$ and $\mathbf{T}_2$ to $\mathcal{I}$
  $Inst_{\mathbf{T}}(n) = Inst_{\mathbf{T}_1}(n_1) \cup Inst_{\mathbf{T}_2}(n_2)$;
end while
return $\mathcal{I}$

Algorithm 2 MAX-PROOF-TREE($V, \mathcal{Q}, U$).

Compute MAX-PROOF-FOREST($\mathcal{Q}, U$).
return The unique tree $\mathbf{T}$, with $V(body_{\mathcal{Q}}) \subseteq Inst_{\mathbf{T}}(n)$, where n is the root of $\mathbf{T}$.

Recall that under set semantics $\mathbf{PC\text{-}Trans}^{set}(\mathbf{CQ}^{\neq}, \mathbf{CQ}^{\neq})$ is Π_3^p-complete [5]. In the remainder of this section, we obtain the following result:

▶ **Theorem 4.6.** *$\mathbf{PC\text{-}Trans}^{bag}(\mathbf{CQ}^{\neq}, \mathbf{CQ}^{\neq})$ is in* EXPTIME.

We introduce IF-proof-trees as a means for reasoning on implied facts.

▶ **Definition 4.7.** For a query $\mathcal{Q}$ and universe $U \subseteq \mathbf{dom}$, an *IF-proof-tree $\boldsymbol{T}$ for $\mathcal{Q}$ over U* is a binary tree in which all nodes n have an instance $Inst_{\mathbf{T}}(n)$ as label with the following conditions:

1. If n is a leaf, then $Inst_{\mathbf{T}}(n) = V(body_{\mathcal{Q}})$ for some valuation V for $\mathcal{Q}$ over U;
2. If n is an intermediate node with children n_1 and n_2, then $Inst_{\mathbf{T}}(n) = Inst_{\mathbf{T}}(n_1) \cup Inst_{\mathbf{T}}(n_2)$, and some valuation V for $\mathcal{Q}$ over U exists with $V(body_{\mathcal{Q}}) \subseteq Inst_{\mathbf{T}}(n_1) \cap Inst_{\mathbf{T}}(n_2)$.

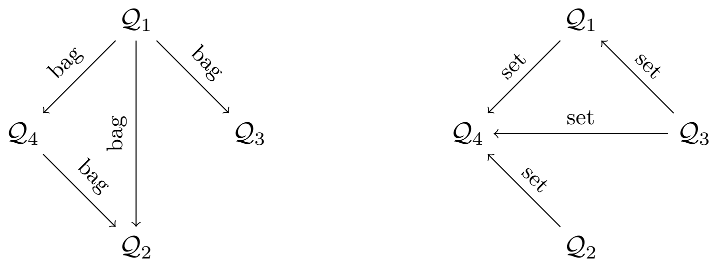
In the next lemma, we relate IF-proof-trees and bag-parallel-correct distribution policies. In particular, the lemma says that all facts occurring together in an IF-proof-tree for a given query have to be assigned to exactly one compute node by every distribution policy that is bag-parallel-correct for that query.

▶ **Lemma 4.8.** *Let $\mathcal{Q} \in \mathbf{CQ}^{\neq}$ and $\boldsymbol{T}$ an IF-proof-tree over universe U'. For every distribution policy $\boldsymbol{P} = (U, rfacts_{\boldsymbol{P}})$ with $U' \subseteq U$ (over some network $\mathcal{N}$) that is bag-parallel-correct for $\mathcal{Q}$, there is exactly one node $\kappa \in \mathcal{N}$, with $Inst_{\boldsymbol{T}}(n) \subseteq rfacts_{\boldsymbol{P}}(\kappa)$, for every n in $\boldsymbol{T}$.*

Algorithm 1 is a procedure that constructs all maximal IF-proof-trees. We notice that at each point during the evaluation of MAX-PROOF-FOREST($\mathcal{Q}, U$), all trees in $\mathcal{I}$ are valid IF-proof-trees for $\mathcal{Q}$ and U, by construction. In particular, the output of Algorithm 1 contains for every valuation V a unique tree with $V(body_{\mathcal{Q}}) \subseteq Inst_{\mathbf{T}}(n)$. Indeed, if two such trees would exist, they would have been combined into a new tree by construction. Algorithm 2 then selects the unique tree w.r.t. a given valuation. We notice that MAX-PROOF-TREE($V, \mathcal{Q}, U$) is well-defined, since, if V is a valuation for $\mathcal{Q}$ over U, then the desired tree $\mathbf{T}$ indeed exists.

The next lemma shows that MAX-PROOF-TREE($V, \mathcal{Q}, U$) computes precisely the facts that are implied by V and $\mathcal{Q}$.

▶ **Lemma 4.9.** *For a query $\mathcal{Q}$ and valuation V for $\mathcal{Q}$, $\boldsymbol{f} \in ImpFacts(V, \mathcal{Q})$ if and only if $\boldsymbol{f} \in Inst_{\boldsymbol{T}}(n)$, with n being the root of $\boldsymbol{T}$ =*MAX-PROOF-TREE($V, \mathcal{Q}, U$).



(a) Bag-parallel-correctness transfer. **(b)** Set-parallel-correctness transfer.

Figure 1 Relationship between the queries of Section 4.3 with respect to (a) bag-parallel-correctness transfer and (b) set-parallel-correctness transfer.

Observe that when U is finite, MAX-PROOF-TREE$(V, \mathcal{Q}, U)$ runs in time exponential in the size of $\mathcal{Q}$ and U. The next lemma says that we can restrict attention to finite universes of size bounded by the number of variables in the queries.

▶ **Lemma 4.10.** *Let $\mathcal{Q}, \mathcal{Q}' \in \mathbf{CQ}^{\neq}$ and $\mathbf{dom}_k = \{1, \ldots, k\}$ be a subset of $\mathbf{dom}$, where $k = max(|\mathit{Vars}(\mathcal{Q})|, |\mathit{Vars}(\mathcal{Q}')|)$. The following conditions are equivalent:*

(1) *For each valuation V' for $\mathcal{Q}'$ over $U \subseteq \mathbf{dom}$, there exists a valuation V for $\mathcal{Q}$ over U such that $V(\mathit{body}_{\mathcal{Q}}) \subseteq V'(\mathit{body}_{\mathcal{Q}'}) \subseteq \mathit{ImpFacts}(V, \mathcal{Q})$.*

(2) *For each valuation W' for $\mathcal{Q}'$ over $U_k \subseteq \mathbf{dom}_k$, there exists a valuation W for $\mathcal{Q}$ over U_k such that $W(\mathit{body}_{\mathcal{Q}}) \subseteq W'(\mathit{body}_{\mathcal{Q}'}) \subseteq \mathit{ImpFacts}(W, \mathcal{Q})$.*

We are now ready to prove Theorem 4.6.

Proof. *(of Theorem 4.6)* The proof is by a naive verification of condition (2) of Lemma 4.10. More specifically, for every universe $U \subseteq \mathbf{dom}_k$ and every valuation V for $\mathcal{Q}$ over U, we compute $\mathit{ImpFacts}(V, \mathcal{Q})$ through MAX-PROOF-TREE$(V, \mathcal{Q}, U)$ (cf. Lemma 4.9). Then, for every valuation V' for $\mathcal{Q}'$ over U and every valuation V for $\mathcal{Q}$ over U we test condition $V(\mathit{body}_{\mathcal{Q}}) \subseteq V'(\mathit{body}_{\mathcal{Q}'}) \subseteq \mathit{ImpFacts}(V, Q)$. If for some V' no V is found that satisfies the condition, then the algorithm returns *false*, otherwise it returns *true*.

Correctness of the algorithm follows directly from Lemma 4.10 and Lemma 4.5. It remains to show that this algorithm proceeds in exponential time in the size of $\mathcal{Q}$ and $\mathcal{Q}'$. For this, we recall that $\mathbf{dom}_k$ is linear in $\mathcal{Q}$ and $\mathcal{Q}'$ by construction, and thus that there are only exponentially many universes $U \subseteq \mathbf{dom}_k$ (w.r.t $\mathcal{Q}$ and $\mathcal{Q}'$). The set of implied facts for a given V and $\mathcal{Q}$, restricted to U, is computable in exponential time and itself is of at most exponential size. Since only exponentially many valuations for $\mathcal{Q}$ and $\mathcal{Q}'$ exist over U, and the test condition itself proceeds in a linear run over the set of implied facts, the result follows. ◀

4.3 Relationship between transferability under set and bag semantics

We argue that set-parallel-correctness transfer is orthogonal to bag-parallel-correctness transfer. Indeed, consider the following queries:

$\mathcal{Q}_1 : H() \leftarrow R(x, y), R(z, w).$

$\mathcal{Q}_2 : H() \leftarrow R(x, x), R(y, y), R(z, z), x \neq y, y \neq z, x \neq z.$

$\mathcal{Q}_3 : H() \leftarrow R(x, y), R(x, z), y \neq z.$

$\mathcal{Q}_4 : H() \leftarrow R(x, y), R(y, z), R(x, x).$

Figure 1 shows the directions in which set-parallel-correctness transfer and bag-parallel-correctness transfer hold. In particular, when an edge is missing, there is no set- or bag-parallel-correctness transfer between the two queries.

The next lemma follows directly from Theorem 3.8.

▶ **Lemma 4.11.** *For strongly minimal queries $\mathcal{Q}, \mathcal{Q}' \in \mathbf{CQ}^{\neq}$ and non-replicating distribution policies, we have that $\mathcal{Q} \xrightarrow{bag} \mathcal{Q}'$ if and only if $\mathcal{Q} \xrightarrow{set} \mathcal{Q}'$.*

5 Modifying the distribution model

As already hinted upon in the Introduction, the Highlander Lemma of Section 3 implies that the space of valuations for a conjunctive query should be perfectly partitioned over all compute nodes. That is, every valuation should occur in exactly one compute node. We next give a simple example query for which the distribution policies that are bag-parallel-correct for it, have to map all facts to a single node.

▶ **Example 5.1.** Consider the query $\mathcal{Q} : H(x, z) \leftarrow R(x, y), R(y, z)$. We argue that distribution policies that map all facts to a single node are the only distribution policies that are bag-parallel-correct. Indeed, let $\boldsymbol{P}$ be a distribution policy that is bag-parallel-correct for $\mathcal{Q}$. Assume $R(a, a) \in \mathit{rfacts}_{\boldsymbol{P}}(\kappa)$ for some node κ. Then, the valuation $\{x \mapsto a, y \mapsto a, z \mapsto b\}$ (for every b) together with Lemma 3.4, implies that every fact of the form $R(a, b)$ for any b should belong to $\mathit{rfacts}_{\boldsymbol{P}}(\kappa)$ as well. Furthermore, the valuation $\{x \mapsto a, y \mapsto b, z \mapsto c\}$ (for every b and c) together with Lemma 3.4, implies that every fact of the form $R(b, c)$ for any b and any c should belong to $\mathit{rfacts}_{\boldsymbol{P}}(\kappa)$ as well. Consequently, $\boldsymbol{P}$, to be bag-parallel-correct for $\mathcal{Q}$, maps all facts to node κ. ◀

The previous example shows that there are queries where the demand for bag-parallel-correctness effectively prohibits parallel computation. We note that this is not the case for all queries. See for instance Example 4.3.

In this section, we consider the setting of *ordered networks* where every compute node is assigned a number and for every valuation only the node with the smallest number containing all facts required for that valuation can contribute to the query result. While both settings do not differ under set semantics, the new setting is more natural for bag semantics and alleviates the problem put forward in Example 5.1.

We associate a total order $<_{\mathcal{N}}$ to every network $\mathcal{N}$. We refer to these networks as *ordered networks*. The definition of a distribution policy $\boldsymbol{P} = (U, \mathit{rfacts}_{\boldsymbol{P}})$ seamlessly carries over to ordered networks. Let $\mathcal{Q}$ be a query and V be a valuation over U for $\mathcal{Q}$. Then, we say that a node $\kappa \in \mathcal{N}$ is *responsible for V (of $\mathcal{Q}$)* if $V(\mathit{body}_{\mathcal{Q}}) \subseteq \mathit{rfacts}_{\boldsymbol{P}}(\kappa)$ and there is no node $\kappa' \in \mathcal{N}$ with $\kappa' <_{\mathcal{N}} \kappa$ and $V(\mathit{body}_{\mathcal{Q}}) \subseteq \mathit{rfacts}_{\boldsymbol{P}}(\kappa')$. Intuitively, the node responsible for a valuation V is the smallest node in the ordered network containing all the facts for $V(\mathit{body}_{\mathcal{Q}})$.

We redefine the one-round distributed evaluation induced by $\boldsymbol{P}$ and $<_{\mathcal{N}}$ as follows:

$$[\mathcal{Q}, \boldsymbol{P}, <_{\mathcal{N}}](I) = \biguplus_{\kappa \in \mathcal{N}, V \in \mathcal{V}_\kappa} [\mathcal{Q}, V](\mathit{loc\text{-}inst}_{\boldsymbol{P},I}(\kappa))$$

with $\mathcal{V}_\kappa$ the set of valuations for which κ is responsible.

The notions of set- and bag-parallel-correctness carry over directly to the setting of ordered networks. Notice that under set-semantics it does not matter whether the ordering of nodes is taken into account.

▶ **Proposition 5.2.** *For each query $\mathcal{Q}$, distribution policy $\boldsymbol{P}$, and ordered network $(\mathcal{N}, <_{\mathcal{N}})$, the following hold for all instances I:*

1. $[\mathcal{Q}, \boldsymbol{P}, <_{\mathcal{N}}](I) \subseteq [\mathcal{Q}, \boldsymbol{P}](I)$;
2. $[\mathcal{Q}, \boldsymbol{P}, <_{\mathcal{N}}](I) \subseteq \mathcal{Q}(I)$; *and,*
3. $\mathit{Facts}([\mathcal{Q}, \boldsymbol{P}](I)) = \mathit{Facts}([\mathcal{Q}, \boldsymbol{P}, <_{\mathcal{N}}](I))$;

In particular, Proposition 5.2(3) implies that Theorem 3.2 and Lemma 3.3 carry over to ordered networks. The next lemma provides characterizations of bag-parallel-correctness and transferability over ordered networks.

▶ **Lemma 5.3.** *Let $\mathcal{Q}$ and $\mathcal{Q}'$ be in $\mathbf{CQ}^{\neq}$. Let $\boldsymbol{P} = (U, \mathit{rfacts}_{\boldsymbol{P}})$ be a distribution policy over an ordered network $\mathcal{N}$. Then the following characterizations hold true:*

1. *$\mathcal{Q}$ is bag-parallel-correct under $\boldsymbol{P}$ if and only if for every valuation V for $\mathcal{Q}$ over U there is a node κ with $V(\mathit{body}_{\mathcal{Q}}) \subseteq \mathit{rfacts}_{\boldsymbol{P}}(\kappa)$; and,*
2. *bag-parallel-correctness transfers from $\mathcal{Q}$ to $\mathcal{Q}'$ over ordered networks if and only if for each valuation V' for $\mathcal{Q}'$ over a universe U' there is a valuation V for $\mathcal{Q}$ over U' such that $V'(\mathit{body}_{\mathcal{Q}'}) \subseteq V(\mathit{body}_{\mathcal{Q}})$.*

Notice the similarity with Lemma 3.3 and Lemma 4.2. In particular, the inclusion between $V(\mathit{body}_{\mathcal{Q}})$ and $V'(\mathit{body}_{\mathcal{Q}'})$ now is in the same direction as in Lemma 4.2. The only difference is that in the above lemma *all* valuations are considered rather than only the minimal ones. The latter is reflected in the complexity of the associated decision problems.

We formally define the respective decision problems. By $\mathcal{C}$ and $\mathcal{C}'$ we denote query classes, by $\mathcal{P}$ a class of distribution policies.

	$\mathbf{PC}^{bag}{}_{<_{\mathcal{N}}}(\mathcal{C}, \mathcal{P})$
Input:	Query $\mathcal{Q} \in \mathcal{C}$, distribution policy $\boldsymbol{P} \in \mathcal{P}$
Question:	Is $\mathcal{Q}$ bag-parallel-correct under $\boldsymbol{P}$?

	$\mathbf{PC\text{-}Trans}^{bag}{}_{<_{\mathcal{N}}}(\mathcal{C}, \mathcal{C}')$
Input:	Query $\mathcal{Q} \in \mathcal{C}$, query $\mathcal{Q}' \in \mathcal{C}'$
Question:	Does bag-parallel-correctness transfer from $\mathcal{Q}$ to $\mathcal{Q}'$?

Using the characterizations in Lemma 5.3, we obtain the following results.

▶ **Theorem 5.4.** **1.** *$\mathbf{PC}^{bag}{}_{<_{\mathcal{N}}}(\mathbf{CQ}, \mathcal{P}_{\mathit{fin}})$ is* CONP*-hard and $\mathbf{PC}^{bag}{}_{<_{\mathcal{N}}}(\mathbf{CQ}^{\neq}, \mathcal{P})$ is in* CONP *for all $\mathcal{P} \in \{\mathcal{P}_{\mathit{fin}}\} \cup \mathfrak{P}_{det}$; and*

2. *$\mathbf{PC\text{-}Trans}^{bag}{}_{<_{\mathcal{N}}}(\mathbf{CQ}^{\neq}, \mathbf{CQ}^{\neq})$ and $\mathbf{PC\text{-}Trans}^{bag}{}_{<_{\mathcal{N}}}(\mathbf{CQ}^{\neq}, \mathbf{CQ})$ are Π_2^p-complete; and*
3. *$\mathbf{PC\text{-}Trans}^{bag}{}_{<_{\mathcal{N}}}(\mathbf{CQ}, \mathbf{CQ}^{\neq})$ and $\mathbf{PC\text{-}Trans}^{bag}{}_{<_{\mathcal{N}}}(\mathbf{CQ}, \mathbf{CQ})$ are* NP*-complete.*

Proof sketch. (1) We first argue that $\mathbf{PC}^{bag}{}_{<_{\mathcal{N}}}(\mathbf{CQ}^{\neq}, \mathcal{P})$ is in CONP for $\mathcal{P} \in \{\mathcal{P}_{\mathit{fin}}\} \cup \mathfrak{P}_{det}$. The required algorithm follows from Lemma 5.3(1). It suffices to guess a valuation V and a node κ and verify that $V(\mathit{body}_{\mathcal{Q}}) \not\subseteq \mathit{rfacts}_{\boldsymbol{P}}(\kappa)$ to check whether $\mathcal{Q}$ is not bag-parallel-correct under a given distribution policy $\boldsymbol{P}$.

To show that $\mathbf{PC}^{bag}{}_{<_{\mathcal{N}}}(\mathbf{CQ}, \mathcal{P}_{\mathit{fin}})$ is CONP-hard, we use a reduction from the problem that asks whether a given graph is *not* 3-colorable. Let G be an arbitrary undirected graph with n edges. We construct a query $\mathcal{Q}$ and policy $\boldsymbol{P}$ over a network with n nodes and a universe $U = \{r, g, b\}$ as follows: For every edge $e = (u, v)$ in G, we add the atom $E_e(x_u, x_v)$ to $\mathit{body}_{\mathcal{Q}}$ and define $\mathit{rfacts}_{\boldsymbol{P}}(e) = \{\boldsymbol{f} \mid \boldsymbol{f} \in \mathit{Facts}(\{E_i\}, U), e \neq i\} \cup \{E_e(r, r), E_e(g, g), E_e(b, b)\}$. Intuitively, each valuation for $\mathcal{Q}$ corresponds to a coloring of G, and only valuations related to an invalid coloring are supported by at least one node.

(2) The algorithm to show that $\textbf{PC-Trans}^{bag}{}_{<_{\mathcal{N}}}(\mathbf{CQ}^{\neq}, \mathbf{CQ}^{\neq})$ is in Π_2^p follows from Lemma 5.3(2). The lower bound is by a non-trivial reduction from the quantified boolean satisfiability problem for the respective level of the hierarchy that is inspired by a technique used in [17].

(3) It can be shown that for a CQ $\mathcal{Q}$, bag-parallel-correctness transfers from $\mathcal{Q}$ to $\mathcal{Q}'$ over ordered networks if and only if a mapping θ for $\mathcal{Q}$ over $adom(body_{\mathcal{Q}'})$ exists such that $body_{\mathcal{Q}'} \subseteq \theta(body_{\mathcal{Q}})$. The required algorithm to show that $\textbf{PC-Trans}^{bag}{}_{<_{\mathcal{N}}}(\mathbf{CQ}, \mathbf{CQ}^{\neq})$ is in NP now follows.

To prove NP-hardness, we provide a reduction from graph 3-colorability. Let G be an arbitrary graph with n edges. We first introduce the following sets of atoms:

$$\texttt{invalidE} = \{E_i(x_i, x_i), E_i(y_i, y_i), E_i(z_i, z_i) \mid i \in [n]\}.$$
$$\texttt{surplusE} = \{E_i(\_,\_), E_i(\_,\_), E_i(\_,\_), E_i(\_,\_), E_i(\_,\_) \mid i \in [n]\ \}.$$

We now define $\mathcal{Q}$ and $\mathcal{Q}'$ as follows:

$$body_{\mathcal{Q}} = \{E_i(x_u, x_v) \mid E(u,v) \in G \text{ having label } i\} \cup \texttt{invalidE} \cup \texttt{surplusE},$$
$$body_{\mathcal{Q}'} = \{E_i(x,y) \mid i \in [n] \text{ and } x, y \in \{x_r, x_g, x_b\}\}.$$

Intuitively, $body_{\mathcal{Q}'} \subseteq \theta(body_{\mathcal{Q}})$ implies that for every edge all colorings can be partitioned into three sets: one valid coloring that participates in the 3-coloring of the graph; the invalid colorings; and, the rest or the surplus of the colorings. ◀

6 Discussion

In this paper, we revisited the framework of [5] under bag semantics. The latter represents a more accurate semantics for real world queries and is a necessary step towards aggregate queries. We obtained semantical characterizations for parallel-correctness as well as transferability under bag semantics. For bag-parallel-correctness we provide tight complexity bounds whereas for transferability we provide an upper bound in EXPTIME. In addition, we show correspondences and incomparabilities with the analog problems under set semantics. We also introduced an ordered network setting that could be more natural for capturing bag semantics and in this setting obtained tight complexity bounds for both decision problems. We mention that all our results can be naturally extended to unions of conjunctive queries. The latter does not need any additional ideas but clutters notation.

There are quite a number of directions for follow-up work. We did not prove a lower bound for transfer of bag-parallel-correctness. Actually, we suspect the upper bound can be improved by coming up with a more efficient algorithm to compute the set of implied facts.

A motivation for the ordered model presented in Section 5 is that bag-parallel-correctness under the previous model can prohibit parallelization. Indeed, Example 5.1 shows a query that can not be parallelized while retaining bag-parallel-correctness. A natural question is whether this class of queries for which no efficient policy exists can be characterized.

Whereas the focus in this paper is on set and bag semantics, it could be interesting to consider parallel-correctness and parallel-correctness transfer under bag-set [7] or combined semantics [8]. Similarly, another direction of future work would be to consider parallel-correctness in the context of aggregate operators.

References

1 Foto N. Afrati, Manas R. Joglekar, Christopher Ré, Semih Salihoglu, and Jeffrey D. Ullman. GYM: A multiround distributed join algorithm. In *ICDT*, pages 4:1–4:18, 2017.

2 Foto N. Afrati, Anish Das Sarma, Semih Salihoglu, and Jeffrey D. Ullman. Upper and lower bounds on the cost of a map-reduce computation. *PVLDB*, 6(4):277–288, 2013.

3 Foto N. Afrati and Jeffrey D. Ullman. Optimizing joins in a map-reduce environment. In *EDBT*, pages 99–110, 2010.

4 Tom J. Ameloot, Gaetano Geck, Bas Ketsman, Frank Neven, and Thomas Schwentick. Data partitioning for single-round multi-join evaluation in massively parallel systems. *SIGMOD Record*, 45(1):33–40, 2016.

5 Tom J. Ameloot, Gaetano Geck, Bas Ketsman, Frank Neven, and Thomas Schwentick. Parallel-correctness and transferability for conjunctive queries. *J. ACM*, 64(5):36:1–36:38, 2017.

6 Paul Beame, Paraschos Koutris, and Dan Suciu. Skew in parallel query processing. In *PODS*, pages 212–223. ACM, 2014.

7 Surajit Chaudhuri and Moshe Y. Vardi. Optimization of *Real* conjunctive queries. In *PODS*, pages 59–70. ACM Press, 1993.

8 Sara Cohen. Equivalence of queries that are sensitive to multiplicities. *VLDB J.*, 18(3):765–785, 2009.

9 Sumit Ganguly, Abraham Silberschatz, and Shalom Tsur. A framework for the parallel processing of datalog queries. In *SIGMOD*, pages 143–152. ACM Press, 1990.

10 Gaetano Geck, Bas Ketsman, Frank Neven, and Thomas Schwentick. Parallel-correctness and containment for conjunctive queries with union and negation. In *ICDT*, pages 9:1–9:17, 2016.

11 Hadoop. URL: `https://hadoop.apache.org/`.

12 Bas Ketsman, Aws Albarghouthi, and Paraschos Koutris. Distribution policies for datalog. In *ICDT*, pages 17:1–17:22, 2018.

13 Bas Ketsman and Dan Suciu. A worst-case optimal multi-round algorithm for parallel computation of conjunctive queries. In *PODS*, pages 417–428, 2017.

14 Paraschos Koutris and Dan Suciu. Parallel evaluation of conjunctive queries. In *PODS*, pages 223–234, 2011.

15 Rimma Nehme and Nicolas Bruno. Automated partitioning design in parallel database systems. In *SIGMOD*, pages 1137–1148. ACM, 2011.

16 Jun Rao, Chun Zhang, Nimrod Megiddo, and Guy Lohman. Automating physical database design in a parallel database. In *SIGMOD*, pages 558–569, 2002.

17 Ron van der Meyden. The complexity of querying indefinite data about linearly ordered domains. In *PODS*, pages 331–345, 1992.

18 Reynold S. Xin, Josh Rosen, Matei Zaharia, Michael J. Franklin, Scott Shenker, and Ion Stoica. Shark: SQL and rich analytics at scale. In *SIGMOD*, pages 13–24, 2013.