Parallel-Correctness and Transferability for Conjunctive Queries under Bag Semantics

Peer-reviewed author version

KETSMAN, Bas; NEVEN, Frank & VANDEVOORT, Brecht (2025)
Parallel-Correctness and Transferability for Conjunctive Queries under Bag Semantics. In: Acm Transactions on Computational Logic, 26 (2) (Art N° 7).

DOI: 10.1145/3712291
Handle: http://hdl.handle.net/1942/45425

Parallel-Correctness and Transferability for Conjunctive Queries under Bag Semantics

BAS KETSMAN, Vrije Universiteit Brussel, Belgium
FRANK NEVEN and BRECHT VANDEVOORT, Hasselt University, Belgium

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.

CCS Concepts: • **Information systems** → **Query languages**; **Parallel and distributed DBMSs**.

Additional Key Words and Phrases: Conjunctive queries, distributed evaluation, bag semantics

1 INTRODUCTION

The rise of parallel data management systems like, for instance, Spark [21] and Hadoop [12], 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, 15, 16]). 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.

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 Q and Q', can we infer from the fact that Q is computed correctly under the current distribution policy, that 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. Parallel-correctness has also been studied in the context of non-oblivious distribution policies. Indeed, Geck et al. [11] introduce a declarative framework for expressing distribution constraints, like co-partitioning constraints, and study their implication problem. The obtained results yield bounds on deciding parallel-correctness for conjunctive queries in the presence of such distribution constraints. Furthermore, Sundarmurthy et al. [19] study parallel-correctness for co-hash distribution schemes.

Finally, Ketsman, Albarghouthi and Koutris [13] 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., [17, 18]). In this setting, partitionings are thus instance dependent and not known in advance.

We say that parallel-correctness transfers from a query Q to a query Q' when Q' is parallel-correct under every distribution policy $\boldsymbol{P}$ under which Q is parallel-correct. We prove the Sandwich Lemma that provides a semantic 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 semantic 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 semantic 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 semantic 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.

The current paper is an extended version of [14] featuring all the proofs, more examples, and a discussion of the relationship with query containment.

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 $Facts(\mathcal{D}, U)$ 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 $Facts(\mathcal{D}, \mathbf{dom})$ as $Facts(\mathcal{D})$.

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 F. That is, $(\boldsymbol{f}, m) \in F$ and $(\boldsymbol{f}', m') \in F$ 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 $Facts(F)$ and the multiplicity of a fact $\boldsymbol{f}$ in the bag F by $mul_F(\boldsymbol{f})$. For convenience, we abuse notation and extend $mul_F(\boldsymbol{f})$ to arbitrary facts by setting $mul_F(\boldsymbol{f}) = 0$ when $\boldsymbol{f} \notin Facts(F)$. 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 $H = F \cup G$, is defined as $Facts(F) \cup Facts(G)$

and $mul_H(f) = mul_F(f) + mul_G(f)$ for each fact $f \in Facts(H)$. Furthermore, F is a *subbag* of G, denoted $F \subseteq G$, if $mul_F(f) \leq mul_G(f)$ for each fact $f \in Facts(F)$. By $|F|$, we denote the number of facts in F, that is, $\sum_{f \in Facts(F)} mul_F(f)$.

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

A *query* 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 Q is *monotone* if $Q(I') \subseteq Q(I)$ 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 $\mathbf{x} = (x_1, \dots, x_k)$ a tuple of variables in **var** with $k = ar(R)$.

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

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

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 $head_Q$, to the set $\{R_1(\mathbf{y_1}), \dots, R_m(\mathbf{y_m})\}$ as $body_Q$ and to the set of all variables occurring in Q as $vars(Q)$.

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 Q is *full* if every variable occurring in Q appears in the head atom.

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

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

$$m = \prod_{f \in V(body_Q)} mul_I(f).$$

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

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

with $\mathcal{V}$ the set containing all valuations that satisfy 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* $\mathbf{P} = (U, rfacts_{\mathbf{P}})$ 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 $\boldsymbol{f} \in rfacts_{\boldsymbol{P}}(\kappa)$.

Example 2.1. For an example of a distribution policy over database schema $\mathcal{D} = \{R^{(2)}, S^{(2)}\}$ and network $\mathcal{N} = \{\kappa_1, \kappa_2\}$, consider $\boldsymbol{P} = (U, rfacts_{\boldsymbol{P}})$ with $U = \{a, b\}$,

$$rfacts_{\boldsymbol{P}}(\kappa_1) = \{R(a,b), S(b,a), R(a,a), R(b,b), S(a,a), S(b,b)\},$$

and

$$rfacts_{\boldsymbol{P}}(\kappa_2) = \{R(b,a), S(a,b), R(a,a), R(b,b), S(a,a), S(b,b)\}.$$

Notice that in this example the facts $F = \{R(a,a), R(b,b), S(a,a), S(b,b)\}$ are assigned by $\boldsymbol{P}$ to both nodes.

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.

Example 2.2. Given instance $I = \{(R(a,b), 1), (R(b,a), 2), (S(a,b), 2)\}$ and distribution policy $\boldsymbol{P}$ from Example 2.1,

$$loc\text{-}inst_{\boldsymbol{P},I}(\kappa_1) = \{(R(a,b), 1)\}, \text{and}$$
$$loc\text{-}inst_{\boldsymbol{P},I}(\kappa_2) = \{(R(b,a), 2), (S(a,b), 2)\}.$$

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 Q is evaluated at each node κ separately, after which the bag union of all results is taken:

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

Example 2.3. Taking as query Q, $O(y) \leftarrow R(x,y), S(y,z)$, and distribution policy $\boldsymbol{P}$ and database instance I from Examples 2.1 and 2.2, respectively, we get

$$[Q, \boldsymbol{P}](I) = \{(O(a), 4)\}.$$

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 $\boldsymbol{f} \in rfacts_{\boldsymbol{P}}(\kappa)$ 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 $\boldsymbol{P} = (U, rfacts_{\boldsymbol{P}})$ 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 $rfacts_{\boldsymbol{P}}(\kappa)$ for a given node κ if $\mathcal{A}_P$ has an accepting run of at most $|(\kappa, f)|^k$ steps on input (κ, f). We define $\mathfrak{P}_{nondet}$ as the set $\{\mathcal{P}^k_{nondet} \mid k \geq 2\}$. We remark that each policy in $\mathcal{P}_{fin}$ 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 $\boldsymbol{P} = (U, rfacts_{\boldsymbol{P}})$ 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 $\boldsymbol{f} \in rfacts_{\boldsymbol{P}}(\kappa)$. 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 $Facts(F)$ denotes the set of facts occurring in the bag F.

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

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

Example 3.2. Let $\boldsymbol{P}$, I and $\boldsymbol{Q}$ be as in Examples 2.1, 2.2, and 2.3, respectively. Query Q is bag-parallel-correct on I under $\boldsymbol{P}$, since $Q(I) = \{(O(a), 4)\}$, and therefore also set-parallel-correct on I under $\boldsymbol{P}$, since $Facts(Q(I)) = \{O(a)\} = Facts([\boldsymbol{Q}, \boldsymbol{P}](I))$. Furthermore, it can be easily verified that Q is in fact set-parallel-correct under P (not depending on a specific I). Query Q is *not* bag-parallel-correct under P, as witnessed for example by the instance $I' = \{(R(a, a), 1), (S(a, a), 1)\}$. Indeed, $\boldsymbol{Q}(I') = \{(O(a), 1)\}$ while $[\boldsymbol{Q}, \boldsymbol{P}](I') = \{(O(a), 2)\}$.

We now formally define the decision problems related to parallel-correctness. In the following, 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:

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

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

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

Theorem 3.3 ([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 $\mathcal{C} \in \{\mathrm{CQ}, \mathrm{CQ}^{\neq}\}$ 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 semantic characterization given in the next lemma. To this end, we need the notion of minimal valuations. For Q in $\mathrm{CQ}^{\neq}$, a valuation V is *minimal* if there is *no valuation V'* for Q that derives the same head fact with a strict subset of body facts, that is, such that $V'(body_Q) \subsetneq V(body_Q)$ and $V'(head_Q) = V(head_Q)$. Recall from the definitions that $V(body_Q)$ always refers to a set of facts, regardless of the considered semantics.

Lemma 3.4 ([5]). *Let Q be in $\mathrm{CQ}^{\neq}$. Then Q is set-parallel-correct under distribution policy $\boldsymbol{P} = (U, rfacts_{\boldsymbol{P}})$ if and only if for every minimal valuation V for Q over U, there is a node $\kappa \in \mathcal{N}$ such that $V(body_Q) \subseteq rfacts_{\boldsymbol{P}}(\kappa)$.*

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 3.4. The characterization for bag-parallel-correctness is again related to valuations but is more strict than the condition of Lemma 3.4 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 $Q \in \mathrm{CQ}^{\neq}$ and distribution policy $\boldsymbol{P}$, we say that node κ *supports* valuation V for Q, if $V(body_Q) \subseteq rfacts_{\boldsymbol{P}}(\kappa)$. By $Sup_{\boldsymbol{P}}(Q, V)$, we denote the set of all nodes that support V under $\boldsymbol{P}$.

Lemma 3.5 (Highlander Lemma[1]). *For $Q \in \mathrm{CQ}^{\neq}$ and a distribution policy $\boldsymbol{P} = (U, rfacts_{\boldsymbol{P}})$ over $\mathcal{N}$, Q is bag-parallel-correct under $\boldsymbol{P}$ if and only if $|Sup_{\boldsymbol{P}}(Q, V)| = 1$, for every valuation V for Q.*

Proof. *(If).* Let I be an arbitrary instance. By assumption, for every valuation V for Q there is exactly one node κ_V for which $Sup_{\boldsymbol{P}}(Q, V) = \{\kappa_V\}$. As Q is monotone it follows that $[Q, \boldsymbol{P}](I) \subseteq Q(I)$. It remains to argue $Q(I) \subseteq [Q, \boldsymbol{P}](I)$. Specifically, we show that for every fact $\boldsymbol{f}$,

$$(\boldsymbol{f}, m) \in Q(I) \text{ implies } (\boldsymbol{f}, m') \in [Q, \boldsymbol{P}](I), \text{ with } m = m'. \quad (1)$$

For this, let $\mathcal{V}_{\boldsymbol{f}}$ be the set of all valuations that satisfy Q on I and derive $\boldsymbol{f}$. In other words, $\mathcal{V}_{\boldsymbol{f}}$ contains every valuation V where $V(head_Q) = \boldsymbol{f}$ and $V(body_Q) \subseteq I$. By definition, $m = \sum_{V \in \mathcal{V}_{\boldsymbol{f}}} |[Q, V](I)|$. Then, $(\boldsymbol{f}, m') \in [Q, \boldsymbol{P}](I)$

[1]"There can be only one." https://en.wikipedia.org/wiki/Highlander_(film)

for an $m' > 0$ as $|Sup_{\boldsymbol{P}}(Q, V)| > 0$. Furthermore,

$$m' = \sum_{\kappa \in \mathcal{N}} \sum_{V \in \mathcal{V}_f} |[Q, V](\textit{loc-inst}_{\boldsymbol{P},I}(\kappa))| \tag{2}$$

$$= \sum_{V \in \mathcal{V}_f} |[Q, V](\textit{loc-inst}_{\boldsymbol{P},I}(\kappa_V))| \qquad \text{(as } Sup_{\boldsymbol{P}}(Q, V) = \{\kappa_V\} \text{ for all } V) \tag{3}$$

$$= \sum_{V \in \mathcal{V}_f} |[Q, V](I)| \tag{4}$$

$$= m. \tag{5}$$

Notice that (4) follows as for each compute node κ, either κ has no copy of a fact, or κ has as many copies as there are in instance I. Therefor, if a valuation satisfies Q locally, it will derive as many copies of the head fact as it would derive on instance I. This concludes the proof of (1), and thus of the (if) direction.

(Only-if). Let Q be bag-parallel-correct for $\boldsymbol{P}$ and let V be an arbitrary valuation for Q. We argue that $|Sup_{\boldsymbol{P}}(Q, V)| = 1$.

For this, we first show:

$$\textit{For all valuations } W \textit{ for } Q,\ Sup_{\boldsymbol{P}}(Q, W) \leq 1. \tag{6}$$

For its proof, assume towards a contradiction that for some valuation W for Q, $|Sup_{\boldsymbol{P}}(Q, W)| > 1$. This means that there are at least two distinct nodes $\kappa_1, \kappa_2 \in Sup_{\boldsymbol{P}}(Q, W)$. Let $I = \{(\boldsymbol{g}, 1) \mid \boldsymbol{g} \in W(body_Q)\}$ and $\boldsymbol{f} = W(head_Q)$. We argue that $[Q, \boldsymbol{P}](I)$ derives too many copies of $\boldsymbol{f}$. Indeed, by definition of I it follows that $I = \textit{loc-inst}_{\boldsymbol{P},I}(\kappa_1) = \textit{loc-inst}_{\boldsymbol{P},I}(\kappa_2)$. So, $0 < mul_{Q(I)}(\boldsymbol{f}) = mul_{Q(\textit{loc-inst}_{\boldsymbol{P},I}(\kappa_1))}(\boldsymbol{f}) = mul_{Q(\textit{loc-inst}_{\boldsymbol{P},I}(\kappa_2))}(\boldsymbol{f})$. It follows that $mul_{Q(I)}(\boldsymbol{f}) < mul_{Q(\textit{loc-inst}_{\boldsymbol{P},I}(\kappa_1))}(\boldsymbol{f}) + mul_{Q(\textit{loc-inst}_{\boldsymbol{P},I}(\kappa_2))}(\boldsymbol{f}) \leq mul_{[Q,\boldsymbol{P}](I)}(\boldsymbol{f})$, which is the desired contradiction.

Given (6), it remains to argue that V satisfies Q on at least one node. To this end, let $I = \{(\boldsymbol{g}, 1) \mid \boldsymbol{g} \in V(body_Q)\}$, $\boldsymbol{f} = V(head_Q)$, and let $\mathcal{V}_f$ be the set of all valuations that satisfy Q on I and derive $\boldsymbol{f}$. By construction, $(\boldsymbol{f}, m) \in Q(I)$ for some $m \geq 1$. By assumption, as $\boldsymbol{P}$ is bag-parallel-correct for Q, $Q(I) = [Q, \boldsymbol{P}](I)$. And therefore, $(\boldsymbol{f}, m) \in [Q, \boldsymbol{P}](I)$ with

$$mul_{Q(I)}(\boldsymbol{f}) = mul_{[Q,\boldsymbol{P}](I)}(\boldsymbol{f}). \tag{7}$$

In particular, $|[Q, V](I)| \geq 1$, which implies

$$mul_{Q(I)}(\boldsymbol{f}) > \sum_{W \in \mathcal{V}_f \setminus \{V\}} |[Q, W](I)|. \tag{8}$$

Again, we notice that for each node $\kappa \in \mathcal{N}$, either κ has no copy of a fact, or κ has as many copies as there are in instance I. Thus, if a valuation satisfies Q locally, it will derive as many copies of the head fact as it would derive on instance I. Formally, we have that for all valuations W for Q and all nodes κ in network $\mathcal{N}$ of $\boldsymbol{P}$:

$$|[Q, W](\textit{loc-inst}_{\boldsymbol{P},I}(\kappa))| = \begin{cases} |[Q, W](I)| & \text{if } W(body_Q) \subseteq \textit{loc-inst}_{\boldsymbol{P},I}(\kappa)\text{; and} \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

Now, to argue that $|Sup_{\boldsymbol{P}}(Q, V)| = 1$, the proof proceeds by contradiction. That is, we assume $|Sup_{\boldsymbol{P}}(Q, V)| = 0$. (Recall that (6) already implies $|Sup_{\boldsymbol{P}}(Q, V)| \leq 1$.)

Then,

$$
\begin{aligned}
mul_{[Q,P](I)}(f) &= \sum_{W \in \mathcal{V}_f} \sum_{\kappa \in \mathcal{N}} |[Q,W](loc\text{-}inst_{P,I}(\kappa)| \\
&= \sum_{W \in \mathcal{V}_f \setminus \{V\}} \sum_{\kappa \in \mathcal{N}} |[Q,W](loc\text{-}inst_{P,I}(\kappa)| && \text{(as } |Sup_P(Q,V)| = 0) \\
&\leq \sum_{W \in \mathcal{V}_f \setminus \{V\}} |[Q,W](I)| && \text{(follows from (6) and (9))} \\
&< mul_{Q(I)}(f) && \text{(follows from (8)).}
\end{aligned}
$$

Then, the desired contradiction follows from (7). □

Example 3.6. We argued in Example 3.2 that Q is not bag-parallel-correct for the given distribution policy $\boldsymbol{P}$ by exhibiting a concrete counterexample instance. The same conclusion can be obtained through an application of the Highlander Lemma (Lemma 3.5). Indeed, take the valuation $V = \{x \to a, y \to a, z \to a\}$ for Q which has $Sup_{\boldsymbol{P}}(Q,V) = \{\kappa_1, \kappa_2\}$, and therefore $|Sup_{\boldsymbol{P}}(Q,V)| > 1$.

We next obtain the complexity of deciding bag-parallel-correctness. The upper bound follows rather directly from Lemma 3.5. The lower bound is a reduction from the complement of 3-SAT.

Theorem 3.7. $\mathbf{PC}^{bag}(C, \mathcal{P})$ *is* coNP*-complete for every query class* $C \in \{\mathbf{CQ}, \mathbf{CQ}^{\neq}\}$ *and every policy class* $\mathcal{P} \in \{\mathcal{P}_{fin}\} \cup \mathfrak{P}_{det}$*, even over networks with only two nodes.*

Proof. Since $\mathcal{P}_{fin} \in \mathfrak{P}_{det}$, and $\mathbf{CQ} \subseteq \mathbf{CQ}^{\neq}$, it suffices to show that $\mathbf{PC}^{bag}(\mathbf{CQ}^{\neq}, \mathfrak{P}_{det})$ is in coNP and that $\mathbf{PC}^{bag}(\mathbf{CQ}, \mathcal{P}_{fin})$ is coNP-hard.

(Upper bound). Observe due to Lemma 3.5 that $(Q, \boldsymbol{P}) \notin \mathbf{PC}^{bag}(\mathbf{CQ}^{\neq}, \mathfrak{P}_{det})$ implies the existence of a valuation V for Q over the universe of $\boldsymbol{P}$, where either (i) $|Sup_{\boldsymbol{P}}(Q,V)| = 0$ or (ii) $|Sup_{\boldsymbol{P}}(Q,V)| \geq 2$. To show that $\mathbf{PC}^{bag}(\mathbf{CQ}^{\neq}, \mathfrak{P}_{det})$ is in coNP, it suffices to construct a verification algorithm that tests (i) and (ii) for a given certificate valuation V in polynomial time. Towards this algorithm, recall that $\mathfrak{P}_{det} = (\mathcal{N}, n, \mathcal{A}_P)$, and that V is consistent by definition (cf. Section 2.2). The algorithm keeps track of a counter that is initially set to 0. For every $\kappa \in \mathcal{N}$, the algorithm tests whether $(\kappa, \boldsymbol{f})$ is accepted by $\mathcal{A}_P$ for every $\boldsymbol{f} \in V(body_Q)$. If all tests (for fixed κ) succeed, the counter is increased. At the end, the program rejects if its counter is set to 1 (i.e., (i) and (ii) fail), and accepts otherwise. Since $\mathcal{N}$ is part of the input, and $\mathcal{A}_P$ is a deterministic polynomial time algorithm by definition, it is easy to see that the sketched program has the desired properties.

(Lower bound). The proof is by reduction from the coNP-complete 3-UNSAT problem (i.e., the complement of 3-satisability), which accepts all propositional formulas in 3-CNF that are *not* satisfiable. Specifically, ψ is a conjunction of clauses $C_1 \wedge C_2 \wedge \cdots \wedge C_k$, with each clause consisting of three literals $C_j = (\ell_{1_j} \vee \ell_{2_j} \vee \ell_{3_j})$. Here, a literal ℓ_{i_j} is either a variable $x \in \mathbf{var}$ or a negated variable $\neg x$. Let n denote the number of different variables occurring in ψ.

We construct a query $Q \in \mathbf{CQ}$ and distribution policy $\boldsymbol{P} \in \mathcal{P}_{fin}$ based on ψ and show that $\psi \in$ 3-UNSAT if, and only if, $(Q, \boldsymbol{P}) \in \mathbf{PC}^{bag}(\mathbf{CQ}, \mathcal{P}_{fin})$.

For the construction of Q, we take for every $i \in [1, n]$ a variable x_i and $\overline{x}_i$. Intuitively, x_i represents a variable from ψ while $\overline{x}_i$ represents its negation $\neg x_i$. For convenience, we overload the notation of ℓ_{j_k}: if ℓ_{j_k} represents a negated variable $\neg x_i$, then ℓ_{j_k} denotes the variable $\overline{x}_i$. Now let Q be as follows:

$$head_Q = H(); and,$$
$$body_Q = \{Neg(x_i, \overline{x}_i) \mid i \in [1, n]\} \cup \{C_j(\ell_{j_1}, \ell_{j_2}, \ell_{j_3}) \mid j \in [1, k]\}.$$

Next, we define $\boldsymbol{P}$. Let $\mathbb{B} = \{0, 1\}$. Let $\mathcal{N} = \{\kappa_1, \kappa_2\}$ be a network with two distinct nodes, and let $U = \mathbb{B}$. We construct $\boldsymbol{P}$ as follows:

$$\{(\kappa_1, Neg(\boldsymbol{b})) \mid \boldsymbol{b} \in \mathbb{B}^2\} \cup \{(\kappa_1, C_j(\boldsymbol{b})) \mid \boldsymbol{b} \in \mathbb{B}^3\} \cup$$
$$\{(\kappa_2, Neg(0, 1)), (\kappa_2, Neg(1, 0))\} \cup \{(\kappa_2, C_j(\boldsymbol{b})) \mid \boldsymbol{b} \in \mathbb{B}^3 \setminus \{0, 0, 0\}\}$$

The intuition behind the construction is that every valuation for Q satisfies Q on node κ_1. On node κ_2, only valuations that encode a valid truth assignment for ψ can satisfy Q. It is easy to see that the construction is polynomial. It remains to argue that $\psi \in$ 3-UNSAT if, and only if, $(Q, \boldsymbol{P}) \in \mathbf{PC}^{bag}(\mathrm{CQ}, \mathcal{P}_{fin})$.

(If). Let $\psi \notin$ 3-UNSAT. Then, a satisfying truth assignment β exists for ψ. Let V be the accompanying valuation for Q, that is, $V(x_i) = \beta(x_i)$, and $V(\overline{x}_i) = \neg\beta(x_i)$, for all variables x_i in ψ. We next observe that $V(body_Q) \subseteq rfacts_{\boldsymbol{P}}(\kappa_1)$ and $V(body_Q) \subseteq rfacts_{\boldsymbol{P}}(\kappa_2)$. (Which follows directly from the construction of $\boldsymbol{P}$). Now, it follows from Lemma 3.5 that $(Q, \boldsymbol{P}) \notin \mathbf{PC}^{bag}(\mathrm{CQ}, \mathcal{P}_{fin})$.

(Only-If). Let $\psi \in$ 3-UNSAT. We use Lemma 3.5 to show $(Q, \boldsymbol{P}) \in \mathbf{PC}^{bag}(\mathrm{CQ}, \mathcal{P}_{fin})$. For this, we first observe that $|Sup_{\boldsymbol{P}}(Q, V)| \geq 1$ for every valuation V over universe U, since $V(body_Q) \subseteq rfacts_{\boldsymbol{P}}(\kappa_1)$. Second, towards a contradiction, assume $|Sup_{\boldsymbol{P}}(Q, V)| > 1$, which implies $V(body_Q) \subseteq rfacts_{\boldsymbol{P}}(\kappa_2)$, for some V. Now let β be the truth assignment for ψ corresponding to V, that is $\beta(x_i) = V(x_i)$, for every variable $x_i \in \psi$. Then it is easy to observe from the construction that β is a well-defined truth assignment for ψ, and that β evaluates to 1 (because κ_2 accepts only satisfying clauses), which is the desired contradiction. ◻

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 3.4 and Lemma 3.5. A counterexample for the converse is given in Example 3.9.

PROPOSITION 3.8. *Bag-parallel-correctness implies set-parallel-correctness for queries in* $\mathrm{CQ}^{\neq}$*, but not vice-versa.*

Example 3.9. For an example showing that the reverse direction of Proposition 3.8 does not hold, consider query Q: $T(x) \leftarrow R(x)$. Let $\boldsymbol{P} = (U, rfacts_{\boldsymbol{P}})$ be a distribution policy over network $\mathcal{N} = \{\kappa_1, \kappa_2\}$, with $rfacts_{\boldsymbol{P}}(\kappa_1) = rfacts_{\boldsymbol{P}}(\kappa_2) = \{R(a), R(b)\}$, and $U = \{a, b\}$.

We observe that Q has only two valuations under U which in addition are minimal: $V_a = \{x \mapsto a\}$ and $V_b = \{x \mapsto b\}$. Since $Sup_{\boldsymbol{P}}(Q, V_1) = Sup_{\boldsymbol{P}}(Q, V_2) = \{\kappa_1, \kappa_2\}$ it follows immediately from Lemma 3.4 and Lemma 3.5 that Q is set-parallel-correct, but not bag-parallel-correct, under $\boldsymbol{P}$.

Interestingly, we can identify a class of $\mathrm{CQ}^{\neq}$-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 $\mathrm{CQ}^{\neq}$ 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 $\boldsymbol{P} = (U, rfacts_{\boldsymbol{P}})$ over a network $\mathcal{N}$ is *non-replicating* if and only if $rfacts_{\boldsymbol{P}}(\kappa_1) \cap rfacts_{\boldsymbol{P}}(\kappa_2) = \emptyset$ for every pair of nodes $\kappa_1, \kappa_2 \in \mathcal{N}$ with $\kappa_1 \neq \kappa_2$.

Theorem 3.10. *For a strongly minimal query Q in $\mathrm{CQ}^{\neq}$ and a non-replicating distribution policy $\boldsymbol{P}$, Q is bag-parallel-correct under $\boldsymbol{P}$ iff Q is set-parallel-correct under $\boldsymbol{P}$.*

Proof. It follows from Proposition 3.8 that bag-parallel-correctness of Q under $\boldsymbol{P}$ implies set-parallel-correctness. We show the reverse direction through Lemma 3.5. For this, let V be an arbitrary valuation for Q. Since Q is set-parallel-correct under $\boldsymbol{P}$, and V is minimal (due to strong minimality of Q), it follows from Lemma 3.4 that $|Sup_{\boldsymbol{P}}(Q,V)| \geq 1$. Since $\boldsymbol{P}$ is non-replicating, the latter implies $|Sup_{\boldsymbol{P}}(Q,V)| = 1$. □

Notice that in the constructed counterexample from Example 3.9, the query Q is strongly minimal, but $\boldsymbol{P}$ is replicating. In the following example we show that, for Theorem 3.10, the condition that Q is strongly minimal can not be dropped.

Example 3.11. Consider query $Q\colon T(x) \leftarrow R(x), R(y)$, and network $\mathcal{N} = \{\kappa_1, \kappa_2\}$. Let $\boldsymbol{P} = (U, rfacts_{\boldsymbol{P}})$ be a distribution policy over $U = \{a, b\}$ and $\mathcal{N}$, with $rfacts_{\boldsymbol{P}}(\kappa_1) = \{R(a)\}$ and $rfacts_{\boldsymbol{P}}(\kappa_2) = \{R(b)\}$. Notice that $\boldsymbol{P}$ is non-replicating.

We observe that $\boldsymbol{P}$ is set-parallel-correct for Q. Indeed, there are only two minimal valuations for 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 3.4.

For non-minimal valuation $V = \{x \mapsto a, y \mapsto b\}$, we observe that $|Sup_{\boldsymbol{P}}(Q,V)| = \emptyset$. Thus $\boldsymbol{P}$ cannot be bag-parallel-correct for Q (due to Lemma 3.5).

4 TRANSFERABILITY

Parallel-correctness *transfers* from a query Q to a query Q' when Q' is parallel-correct under every distribution policy $\boldsymbol{P}$ under which Q is parallel-correct. This means in particular that query Q' can *always* be evaluated after query 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 Q and Q' over the same input schema, *bag-parallel-correctness transfers from Q to Q'* if Q' is bag-parallel-correct under every distribution policy for which Q is bag-parallel-correct. In this case, we write $Q \xrightarrow{\text{bag}} Q'$. Set-parallel-correctness transferability is defined similarly and denoted by $Q \xrightarrow{\text{set}} Q'$.

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

4.2 Transferability under bag semantics

We start by observing that for Boolean queries $Q, Q' \in \mathrm{CQ}^{\neq}$, $Q \xrightarrow{\text{bag}} Q'$ implies that Q' is set-contained in Q. Indeed, when Q' is not set-contained in Q, there is a valuation V' for Q' with the property that $Q'(V'(body_{Q'})) \neq \emptyset$ and $Q(V'(body_{Q'})) = \emptyset$. But then the distribution policy over a two-node network assigning all facts to one node and precisely the facts $V'(body_{Q'})$ to the other node is trivially bag-parallel-correct for Q while it is not for Q'. This observation can be further generalized by considering for arbitrary conjunctive queries $Q \in \mathrm{CQ}^{\neq}$ their “Booleanisation”, defined as the conjunctive query obtained by removing from Q all variables from its head. In the following lemma, we write $Q \subseteq_B Q'$ for conjunctive queries Q and Q' to denote set-contained of the "Booleanisation" of Q in the "Booleanisation" of Q'.

Lemma 4.3. *For queries $Q, Q' \in \mathrm{CQ}^{\neq}$, $Q \xrightarrow{bag} Q'$ implies $Q' \subseteq_B Q$.*

Proof. The proof is by contraposition where we show that $Q' \not\subseteq_B Q$ implies $Q \overset{\text{bag}}{\not\longrightarrow} Q'$. The former means that there is an instance I such that $|Q'(I)| \neq \emptyset$ and $|Q(I)| = \emptyset$. Now consider a distribution policy over a two-node network with one node, say κ_1, made responsible for all facts (including those in I); and another node, say κ_2, made responsible for only the facts occurring in I. By choice of I, query Q' is not bag-parallel-correct on $\boldsymbol{P}$ (a direct consequence of the Highlander Lemma), while query Q clearly is bag-parallel-correct, implying $Q \overset{\text{bag}}{\not\longrightarrow} Q'$. □

Boolean containment, however, does not imply transferability. Indeed, $Q' \subseteq_B Q$ only ensures for every valuation V' for Q' that there is a valuation V for Q with $V(body_Q) \subseteq V'(body_{Q'})$. More precisely, the latter implies that for every distribution policy for which Q is bag-parallel-correct, every valuation of Q' can be supported by at most one node; but does not guarantee that every valuation of Q' is supported by a node as is illustrated in the following example.

Example 4.4. Consider queries

$$Q : H() \leftarrow R(x, x)$$

and

$$Q' : H() \leftarrow R(x, x), R(x, y), R(y, x), x \neq y.$$

Then $Q' \subseteq_B Q$. We next argue that $Q \overset{\text{bag}}{\not\longrightarrow} Q'$. Take for example the distribution policy $\boldsymbol{P}$ over two nodes κ_1, κ_2 with κ_1 made responsible for all facts of the form $R(a, a)$ and κ_2 made responsible for all facts of the form $R(a, b)$ with $a \neq b$. Since κ_1 supports all valuation of Q, and it is the only node supporting such valuations, Q is clearly bag-parallel-correct for $\boldsymbol{P}$. On the other hand, none of the valuations for Q' is supported by $\boldsymbol{P}$, hence Q' is not bag-parallel-correct for $\boldsymbol{P}$.

We remark that set-contained for queries with inequalities is Π_2^p-complete [20].

The observative reader may wonder if transferability under bag-semantics perhaps also requires set-containment in the other direction, as this would guarantee that every valuation for Q' is supported by *at least* one of the nodes of the network. This property however is unnecessarily strong. 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 4.5. Consider the query $Q : H(x) \leftarrow R(x, y), R(x, z)$. Let $\boldsymbol{P}$ be a distribution policy under which Q is bag-parallel-correct. Assume $R(a, a) \in rfacts_{\boldsymbol{P}}(\kappa)$ for some node κ. Then, by Lemma 3.5, 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 4.6. Let V be a valuation for $Q \in \mathrm{CQ}^{\neq}$. A fact $\boldsymbol{f}$ is *implied* by V w.r.t. Q if for every distribution policy $\boldsymbol{P} = (U, rfacts_{\boldsymbol{P}})$, with $adom(V(body_Q)) \subseteq U$ under which Q is bag-parallel-correct, and for every node κ in the network of $\boldsymbol{P}$: $V(body_Q) \subseteq rfacts_{\boldsymbol{P}}(\kappa)$ implies $\boldsymbol{f} \in rfacts_{\boldsymbol{P}}(\kappa)$. We denote the set of facts implied by V w.r.t. Q by $ImpFacts(V, Q)$.

Notice that *ImpFacts*(V, 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, *ImpFacts*$(V, Q) \subseteq rfacts_{\boldsymbol{P}}(\kappa)$ whenever $V(body_Q) \subseteq rfacts_{\boldsymbol{P}}(\kappa)$ for every distribution policy $\boldsymbol{P}$ under which 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 4.7 (Sandwich lemma). *Bag-parallel-correctness transfers from Q to Q' if and only if for each valuation V' for Q' there is a valuation V for Q such that $V(body_Q) \subseteq V'(body_{Q'}) \subseteq ImpFacts(V, Q)$.*

Proof. *(If).* Let $\boldsymbol{P} = (U, rfacts_{\boldsymbol{P}})$ be an arbitrary distribution policy such that Q is bag-parallel-correct under $\boldsymbol{P}$. Let V' be an arbitrary valuation for Q' over U. We argue that $|Sup_{\boldsymbol{P}}(Q', V')| = 1$ which by Lemma 3.5 implies that Q' is bag-parallel-correct under $\boldsymbol{P}$ as well. By assumption there is a valuation V for Q over U such that $V(body_Q) \subseteq V'(body_{Q'}) \subseteq$ *ImpFacts*(V, Q). Then, by Lemma 3.5, $Sup_{\boldsymbol{P}}(Q, V) = \{\kappa\}$ for some node κ and *ImpFacts*$(V, Q) \subseteq rfacts_{\boldsymbol{P}}(\kappa)$. Therefore, $V'(body_{Q'}) \subseteq rfacts_{\boldsymbol{P}}(\kappa)$. So, $|Sup_{\boldsymbol{P}}(Q', V')| \geq 1$. However, as $V(body_Q) \subseteq rfacts_{\boldsymbol{P}}(\kappa)$ and $Sup_{\boldsymbol{P}}(Q, V) = \{\kappa\}$, $|Sup_{\boldsymbol{P}}(Q', V')| = 1$.

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

Case 1: there is no valuation V with $V(body_Q) \subseteq V'(body_{Q'})$. 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 $rfacts_{\boldsymbol{P}}(\kappa_1) = Facts(\mathcal{D}, U)$ and $rfacts_{\boldsymbol{P}}(\kappa_2) = V'(body_{Q'})$. Then, $Sup_{\boldsymbol{P}}(Q', V') = \{\kappa_1, \kappa_2\}$ and Lemma 3.5 implies that $\boldsymbol{P}$ is not bag-parallel-correct for Q'. In contrast, every valuation for Q is supported only on node κ_1 (as none of them are included in $V'(body_{Q'})$) which implies that $\boldsymbol{P}$ is bag-parallel-correct for Q. We conclude that bag-parallel-correctness does not transfer from Q to Q'.

Case 2: for each valuation V, $V(body_Q) \subseteq V'(body_{Q'})$ implies $V'(body_{Q'}) \nsubseteq ImpFacts(V, Q)$. From the previous case, we can assume the existence of a valuation V with $V(body_Q) \subseteq V'(body_{Q'})$. Then, by definition of *ImpFacts*(V, Q), $V'(body_{Q'}) \nsubseteq ImpFacts(V, Q)$ implies that there must be a policy $\boldsymbol{P}$ (over some network $\mathcal{N}$) such that Q is bag-parallel-correct under $\boldsymbol{P}$ and $\boldsymbol{P}$ has a node κ with $V(body_Q) \subseteq rfacts_{\boldsymbol{P}}(\kappa)$ and $V'(body_{Q'}) \nsubseteq rfacts_{\boldsymbol{P}}(\kappa)$. From Lemma 3.5, it follows that for all other nodes κ', that is $\kappa' \in \mathcal{N} \setminus \{\kappa\}$, $V(body_Q) \nsubseteq rfacts_{\boldsymbol{P}}(\kappa')$, and thus $V'(body_{Q'}) \nsubseteq rfacts_{\boldsymbol{P}}(\kappa')$. Hence, $\boldsymbol{P}$ is not bag-parallel-correct for Q' and, consequently, bag-parallel-correctness does not transfer from Q to Q'. □

Notice that the inclusion between $V(body_Q)$ and $V'(body_{Q'})$ in Lemma 4.7 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 C and C' we denote query classes.

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

Fig. 1. Visual depiction of two IF-proof-trees for query Q of Example 4.5 over U with $\{a, b, c\} \subseteq U$.

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

Theorem 4.8. $\mathbf{PC\text{-}Trans}^{bag}(\mathrm{CQ}^{\neq}, \mathrm{CQ}^{\neq})$ *is in* EXPTIME.

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

Definition 4.9. For a query Q and universe $U \subseteq \mathbf{dom}$, an *IF-proof-tree* $\boldsymbol{T}$ *for* Q *over* U is a binary tree in which all nodes n have an instance $\mathit{Inst}_{\mathbf{T}}(n)$ as label with the following conditions:

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

Two example IF-proof-trees for query Q of Example 4.5 over U with $\{a, b, c\} \subseteq U$ are shown in Figure 1. 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 4.10. *Let* $Q \in \mathrm{CQ}^{\neq}$ *and* $\boldsymbol{T}$ *an IF-proof-tree over universe* U'. *For every distribution policy* $\boldsymbol{P} = (U, \mathit{rfacts}_{\boldsymbol{P}})$ *with* $U' \subseteq U$ *(over some network* $\mathcal{N}$*) that is bag-parallel-correct for* Q*, there is exactly one node* $\kappa \in \mathcal{N}$*, with* $\mathit{Inst}_{\boldsymbol{T}}(n) \subseteq \mathit{rfacts}_{\boldsymbol{P}}(\kappa)$*, for every* n *in* $\boldsymbol{T}$.

Proof. If $\mathbf{T}$ consists of a single leaf-node, the lemma holds straightforwardly by bag-parallel-correctness and Lemma 3.5. Indeed, every valuation for Q over U' is supported by $\boldsymbol{P}$ and by exactly one compute node. Otherwise, the result follows by induction on the depth of $\mathbf{T}$. Specifically, for an intermediate node n, with subtrees $\mathbf{T}_1$ and $\mathbf{T}_2$, where κ_1 and κ_2 are the unique nodes where $\mathit{Inst}_{\mathbf{T}}(n_1) \subseteq \mathit{rfacts}_{\boldsymbol{P}}(\kappa_1)$ and $\mathit{Inst}_{\mathbf{T}}(n_2) \subseteq \mathit{rfacts}_{\boldsymbol{P}}(\kappa_2)$, for all nodes n_1 in $\mathbf{T}_1$ and n_2 in $\mathbf{T}_2$, respectively, we argue that $\kappa_1 = \kappa_2$. Indeed, by definition of IF-proof-tree, a valuation V for Q over U' exists, with $V(\mathit{body}_Q) \subseteq \mathit{Inst}_{\mathbf{T}}(n_1)$ and $V(\mathit{body}_Q) \subseteq \mathit{Inst}_{\mathbf{T}}(n_2)$, then $\kappa_1 = \kappa_2$ follows from Lemma 3.5. □

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(Q, U), all trees in $\mathcal{I}$ are valid IF-proof-trees for Q and U, by construction. In particular, the output of Algorithm 1 contains for every valuation V a unique tree with $V(\mathit{body}_Q) \subseteq \mathit{Inst}_{\mathbf{T}}(n)$. 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, Q, U) is well-defined, since, if V is a valuation for Q over U, then the desired tree $\mathbf{T}$ indeed exists.

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

Algorithm 1 MAX-PROOF-FOREST(Q, U)

Let $\mathcal{I}$ be the set of single-node IF-proof-trees, one for each set $V(body_Q)$, where V is a valuation for Q over U.
while Distinct $\mathbf{T}_1, \mathbf{T}_2 \in \mathcal{I}$ and V for Q over U exist, with $V(body_Q) \subseteq Inst_{\mathbf{T}_1}(n_1) \cap Inst_{\mathbf{T}_2}(n_2)$, 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, Q, U)

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

LEMMA 4.11. *For a query Q and valuation V for Q, $f \in ImpFacts(V, Q)$ if and only if $f \in Inst_{\boldsymbol{T}}(n)$, with n being the root of $\boldsymbol{T}$ =*MAX-PROOF-TREE(V, Q, U).

PROOF. *(If).* The proof follows directly from Lemma 4.10. Indeed, since $V(body_Q) \cup \{f\} \subseteq Inst_{\mathbf{T}}(n)$ for the root n of $\mathbf{T}$, it follows immediately that $f \in ImpFacts(V, Q)$.

(Only-If). Assume $f \in ImpFacts(V, Q)$. Define $\boldsymbol{P} = (adom(V(body_Q)), rfacts_{\boldsymbol{P}})$ as the distribution policy based on the IF-proof-trees in the output $\mathcal{I}$ of MAX-PROOF-FOREST(V, Q, U) as follows. We assume a network $\mathcal{N}$ with exactly one node κ per tree $\mathbf{T}'$ in $\mathcal{I}$ and define $rfacts_{\boldsymbol{P}}(\kappa) = Inst_{\mathbf{T}'}(n')$, with n' being the root of $\mathbf{T}'$. By construction of MAX-PROOF-FOREST(Q, U), every valuation for Q over U is supported by exactly one node under $\boldsymbol{P}$. Thus Lemma 3.5 implies bag-parallel-correctness. Now it follows from the construction of $\boldsymbol{P}$ that $rfacts_{\boldsymbol{P}}(\kappa) = Inst_{\mathbf{T}}(n)$ for the node κ for which $V(body_Q) \subseteq rfacts_{\boldsymbol{P}}(\kappa)$. As $f \in rfacts_{\boldsymbol{P}}(\kappa)$, it then follows that $f \in Inst_{\mathbf{T}}(n)$. □

Observe that when U is finite, MAX-PROOF-TREE(V, Q, U) runs in time exponential in the size of 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 4.12. *Let $Q, Q' \in \mathrm{CQ}^{\neq}$ and $\mathbf{dom}_k = \{1, \ldots, k\}$ be a subset of $\mathbf{dom}$, where $k = max(|Vars(Q)|, |Vars(Q')|)$. The following conditions are equivalent:*

(1) *For each valuation V' for Q' over $U \subseteq \mathbf{dom}$, there exists a valuation V for Q over U such that $V(body_Q) \subseteq V'(body_{Q'}) \subseteq ImpFacts(V, Q)$.*
(2) *For each valuation W' for Q' over $U_k \subseteq \mathbf{dom}_k$, there exists a valuation W for Q over U_k such that $W(body_Q) \subseteq W'(body_{Q'}) \subseteq ImpFacts(W, Q)$.*

PROOF. Direction (1) $\Rightarrow$ (2) follows immediately from $\mathbf{dom}_k \subseteq \mathbf{dom}$ and the observation $adom(V(body_Q)) = adom(V'(body_{Q'}))$. Thus if V is defined over $U_k \subseteq \mathbf{dom}_k$, then V' also is.

We next argue (2) $\Rightarrow$ (1). Therefore, assume condition (2) holds. Let V' be an arbitrary valuation for Q' over some universe $U \subseteq \mathbf{dom}$. By $U' \subseteq U$ we denote the active domain values in $V'(body_{Q'})$. Clearly, $|U'| \leq k$. The argument then relies on the genericity of Q, Q', and $ImpFacts(V, Q)$. More specifically, it relies on the existence of a bijective mapping π from U' to $\{1, 2, \ldots, |U'|\}$. Indeed, by assumption, for valuation $W' = \pi \circ V'$ there is a valuation W for Q over $\mathbf{dom}_k$ as in condition (2). Then, $V = \pi^{-1} \circ W'$ is the desired witness for V' in condition (1). □

We are now ready to prove Theorem 4.8.

Proof. *(of Theorem 4.8)* The proof is by a naive verification of condition (2) of Lemma 4.12. More specifically, for every universe $U \subseteq \mathbf{dom}_k$ and every valuation V for Q over U, we compute *ImpFacts*(V, Q) through MAX-PROOF-TREE(V, Q, U) (cf. Lemma 4.11). Then, for every valuation V' for Q' over U and every valuation V for Q over U we test condition $V(body_Q) \subseteq V'(body_{Q'}) \subseteq ImpFacts(V, Q)$. 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 4.12 and Lemma 4.7. It remains to show that this algorithm proceeds in exponential time in the size of Q and Q'. For this, we recall that $\mathbf{dom}_k$ is linear in Q and Q' by construction, and thus that there are only exponentially many universes $U \subseteq \mathbf{dom}_k$ (w.r.t Q and Q'). The set of implied facts for a given V and Q, restricted to U, is computable in exponential time and itself is of at most exponential size. Since only exponentially many valuations for Q and Q' exist over U, and the test condition itself proceeds in a linear run over the set of implied facts, the result follows. ◻

We do not know if $\mathbf{PC\text{-}Trans}^{bag}(\mathbf{CQ}^{\neq}, \mathbf{CQ}^{\neq})$ is complete for EXPTIME. The strongest lower bound we are currently aware of is the following rather trivial result:

Lemma 4.13. $\mathbf{PC\text{-}Trans}^{bag}(\mathbf{CQ}^{\neq}, \mathbf{CQ}^{\neq})$ *is np-hard.*

Proof. The reduction is from the 3-colorability problem for undirected graphs, which is well-known to be NP-complete. Given an undirected graph G as input to this problem, we can construct two queries Q and Q' and show that bag-parallel-correctness transfers from Q to Q' iff G is 3-colorable.

Both queries are defined over schema $\{E^{(2)}\}$. Query Q' is a Boolean query returning true if for at least some choice of three different colors the E relation contains all pairs of different colors. Formally,

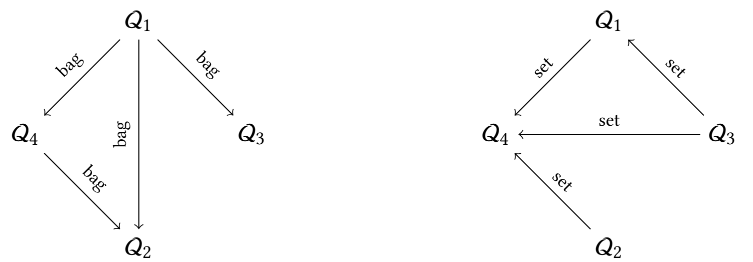
$$Q' : T() \leftarrow E(x, y), E(y, x), E(x, z), E(z, x), E(y, z), E(z, y), x \neq y, x \neq z, y \neq z.$$

Query Q is a Boolean query returning true if Q' returns true *and* there is a mapping f from the nodes in graph G on values in the active domain of the given instance such that for every edge $\{x, y\}$ in G we have that $f(x) \neq f(y)$ and there is a tuple $(f(x), f(y))$ and $(f(y), f(x))$ in the E relation. Formally, $head_Q = T()$, $body_Q = body_{Q'} \cup \{E(u, v) \mid \{v, u\} \in G\} \cup \{E(v, u) \mid \{v, u\} \in G\}$, and Q has a disequality $v \neq u$ for every $\{v, u\} \in G$. In this construction we assume that the variables introduced to represent the nodes of G are all different from the variables used in Q'.

The resulting queries have polynomial size compared to the size of G and can clearly be constructed in polynomial time. Therefore, we only still need to show correctness of the reduction.

Correctness. We first show that if G is 3-colorable, then bag-parallel-correctness transfers from Q to Q'. For this, assume any network and distribution policy that is bag-parallel-correct for Q. We need to show that Q' is also bag-parallel-correct, which (due to Lemma 3.5) is the case if every valuation V' for Q' is supported by precisely one node in the network. To see that the latter is true, we observe that V' for Q' can always be extended to a valuation V for Q with the property that $V(body_Q) = V'(body_{Q'})$. Indeed, this is done by assigning all variables representing nodes of G to a valid 3-coloring making use of the three colors $V(x), V(y), V(z)$, which by construction of Q' are indeed all different. Since bag-parallel-correctness of Q implies that V is supported by precisely one node, and $V(body_Q) = V'(body_{Q'})$, it is immediate that also V is supported by precisely one node, and hence Q is bag-parallel-correct on the given distribution policy.

Second, we show that if G is not 3-colorable, then bag-parallel-correctness does not transfer from Q to Q'. To see this, consider a two-node network, any valuation V' for Q', and a distribution policy making one of the nodes in the



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

Fig. 2. Relationship between the queries of Section 4.3 with respect to (a) bag-parallel-correctness transfer and (b) set-parallel-correctness transfer.

network responsible for precisely the facts $V'(body_{Q'})$ and the other node responsible for all facts (including those in $V'(body_{Q'})$). It follows immediately from Lemma 3.5 and valuation V' that Q' is not bag-parallel-correct for the constructed distribution policy. Query Q however is clearly bag-parallel-correct as all its valuations are supported by at least one of the two nodes, and in case a valuation is supported by both nodes this would directly contradict with G being not 3-colorable. □

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:

$$Q_1 : H() \leftarrow R(x, y), R(z, w).$$

$$Q_2 : H() \leftarrow R(x, x), R(y, y), R(z, z), x \neq y, y \neq z, x \neq z.$$

$$Q_3 : H() \leftarrow R(x, y), R(x, z), y \neq z.$$

$$Q_4 : H() \leftarrow R(x, y), R(y, z), R(x, x).$$

Figure 2 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.

Lemma 4.14. *Set-parallel-correctness transfer and bag-parallel-correctness transfer are orthogonal.*

Proof. We show that Figure 2 is correct. *Case* $Q_1 \overset{set}{\not\longrightarrow} Q_2$, $Q_1 \overset{set}{\not\longrightarrow} Q_3$, $Q_3 \overset{set}{\not\longrightarrow} Q_2$, $Q_4 \overset{set}{\not\longrightarrow} Q_2$, $Q_4 \overset{set}{\not\longrightarrow} Q_3$. The proof follows from Lemma 4.2. Specifically, from the observation that all valuations for Q_2 require three facts, those for Q_3 require two facts, and the minimal valuations for Q_1 and Q_4 require only one fact.

Case $Q_2 \overset{set}{\not\longrightarrow} Q_1$, $Q_2 \overset{set}{\not\longrightarrow} Q_3$, *and* $Q_4 \overset{set}{\not\longrightarrow} Q_1$. Minimal valuations for Q_2 and Q_4 require only facts of the form $R(a, a)$, while minimal valuations exist for Q_1 and Q_3 that require facts of the form $R(a, b)$, with $a \neq b$. The result then follows again from Lemma 4.2.

Case $Q_1 \overset{set}{\longrightarrow} Q_4$, $Q_2 \overset{set}{\longrightarrow} Q_4$, *and* $Q_3 \overset{set}{\longrightarrow} Q_4$. The minimal valuations for Q_4 all require just a single R-fact of the form $R(a, a)$. It is easy to see that valuations $\{x, y, z, w, \mapsto a\}$, $\{x \mapsto a, y \mapsto b, z \mapsto c\}$, and $\{x, y \mapsto a, z \mapsto b\}$ are minimal for Q_1, Q_2, and Q_3 respectively. Then set-parallel-correctness transfer follows from Lemma 4.2.

Case $Q_3 \xrightarrow{set} Q_1$. Minimal valuations for Q_1 all require a single fact of the form $R(a,b)$. Since $\{x \mapsto a, y \mapsto b, z \mapsto c\}$ is a minimal valuation for Q_3, the result again follows from Lemma 4.2.

Case $Q_1 \xrightarrow{bag} Q_2$*,* $Q_1 \xrightarrow{bag} Q_3$*, and* $Q_1 \xrightarrow{bag} Q_4$. We notice that for Q_1 to be bag-parallel-correct over a distribution policy $\boldsymbol{P} = (U, \mathit{rfacts}_{\boldsymbol{P}})$, there is a node κ responsible for all R-facts over U, and $\boldsymbol{f} \in \mathit{rfacts}_{\boldsymbol{P}}(\kappa')$ implies $\kappa' = \kappa$, for all facts $\boldsymbol{f}$ over U with predicate R.

The proof is straightforward. First, we observe that for every combination of facts $R(a,b)$, $R(c,d)$ over U, there is a valuation for Q_1 that requires both. Lemma 3.5 then implies that all these valuations satisfy on some node. Since for every individual fact $R(a,b)$ there is a valuation $\{x, z \mapsto a, y, w \mapsto b\}$ of Q_1 that requires only $R(a,b)$, it follows that all these facts must be mapped on one node, κ.

Case $Q_2 \overset{bag}{\not\longrightarrow} Q_1$*,* $Q_3 \overset{bag}{\not\longrightarrow} Q_1$*,* $Q_4 \overset{bag}{\not\longrightarrow} Q_1$. We observe that there is a valuation for Q_1 that requires only the fact $R(a,b)$, while no valuation for Q_2, Q_3 or Q_4 requires only $R(a,b)$. Therefore, to find a counterexample distribution policy $\boldsymbol{P}(U, \mathit{rfacts}_{\boldsymbol{P}})$, we simply take one that is bag-parallel-correct for Q_2 (or Q_3, Q_4 respectively), with $\{a,b\} \subseteq U$, and then add a new node κ, with $\mathit{rfacts}_{\boldsymbol{P}}(\kappa) = \{R(a,b)\}$. The result follows from Lemma 3.5.

Case $Q_2 \overset{bag}{\not\longrightarrow} Q_3$*, and* $Q_4 \overset{bag}{\not\longrightarrow} Q_3$. For both Q_2 and Q_4 it is easy to see that for valuation $V' = \{x \mapsto a, y \mapsto b, z \mapsto c\}$ of Q_3 the conditions in Lemma 3.5 fail.

Case $Q_2 \overset{bag}{\not\longrightarrow} Q_4$*, and* $Q_3 \overset{bag}{\not\longrightarrow} Q_4$. For Q_2 and Q_3 we observe that for valuation $V' = \{x, y, z \mapsto a\}$ the conditions in Lemma 3.5 fail.

Case $Q_3 \overset{bag}{\not\longrightarrow} Q_2$. The proof is analogous. Here we take $V' = \{x \mapsto a, y \mapsto b, z \mapsto c\}$.

Case $Q_4 \xrightarrow{bag} Q_2$. The result follows from the observations that if $\boldsymbol{P} = (U, \mathit{rfacts}_{\boldsymbol{P}})$ is bag-parallel-correct for Q_4, then there is a node κ where all R-facts over U are mapped on, and as a consequence, that all facts of the form $R(a,a)$, with $a \in U$ are mapped only on κ (due to Lemma 3.5 and existence of valuations for Q_4 that require exactly one such fact $R(a,a)$).

To see why node κ indeed exists, the reasoning is analogous to Example 5.1. □

The next lemma follows directly from Theorem 3.10.

Lemma 4.15. *For strongly minimal queries $Q, Q' \in \mathrm{CQ}^{\neq}$ and non-replicating distribution policies, we have that $Q \xrightarrow{bag} Q'$ if and only if $Q \xrightarrow{set} Q'$.*

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 $Q : H(x,z) \leftarrow R(x,y), R(y,z)$. 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 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 3.5, 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 3.5, 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 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 4.5.

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 Q be a query and V be a valuation over U for Q. Then, we say that a node $\kappa \in \mathcal{N}$ is *responsible for V (of Q)* if $V(\mathit{body}_Q) \subseteq \mathit{rfacts}_{\boldsymbol{P}}(\kappa)$ and there is no node $\kappa' \in \mathcal{N}$ with $\kappa' <_{\mathcal{N}} \kappa$ and $V(\mathit{body}_Q) \subseteq \mathit{rfacts}_{\boldsymbol{P}}(\kappa')$. Intuitively, the node responsible for a valuation V is the smallest node in the ordered network containing all the facts for $V(\mathit{body}_Q)$.

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

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

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

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

Proof. The proof is straightforward. For (1) and (2) we observe that, under the ordered-network semantics, every valuation is applied on at most one node. For (3) we observe that $\mathit{Facts}([Q, \boldsymbol{P}, <_{\mathcal{N}}](I)) \subseteq \mathit{Facts}([Q, \boldsymbol{P}](I))$ due to (1). Further, since all valuations satisfying under the unordered semantics still satisfy on some node, every output fact will still be detected (although frequencies may drop). □

In particular, Proposition 5.2(3) implies that Theorem 3.3 and Lemma 3.4 carry over to ordered networks. The next lemma provides characterizations of bag-parallel-correctness and transferability over ordered networks.

Lemma 5.3. *Let Q and Q' be in $\mathrm{CQ}^{\neq}$. 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) Q is bag-parallel-correct under $\boldsymbol{P}$ if and only if for every valuation V for Q over U there is a node κ with $V(\mathit{body}_Q) \subseteq \mathit{rfacts}_{\boldsymbol{P}}(\kappa)$; and,
(2) bag-parallel-correctness transfers from Q to Q' over ordered networks if and only if for each valuation V' for Q' over a universe U' there is a valuation V for Q over U' such that $V'(\mathit{body}_{Q'}) \subseteq V(\mathit{body}_Q)$.

Proof. *(1).* The proof is straightforward. By definition of query evaluation on ordered networks, it is guaranteed that every valuation V for Q is applied on at most one node. To show bag-parallel-correctness it is thus sufficient (due to Lemma 3.5) to show that every valuation V for Q satisfies on *at least one* node. Indeed, then the lowest of these nodes will apply V.

(2). The result follows directly from (1) and the ordered-network semantics. □

Notice the similarity with Lemma 3.4 and Lemma 4.2. In particular, the inclusion between $V(body_Q)$ and $V'(body_{Q'})$ 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 C and C' we denote query classes, by $\mathcal{P}$ a class of distribution policies.

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

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

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

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

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

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

Proof. (1) We first argue that $\mathbf{PC}^{bag}{}_{<_{\mathcal{N}}}(\mathrm{CQ}^{\neq}, \mathfrak{P}_{det})$ is in coNP. The required algorithm follows immediately from Lemma 5.3(1). Indeed, if a given query Q is not bag-parallel-correct under a given distribution policy $\boldsymbol{P}$, then a valuation V exists such that $V(body_Q) \not\subseteq rfacts_{\boldsymbol{P}}(\kappa)$ for every node κ of the network that $\boldsymbol{P}$ is defined over. By definition of $\mathfrak{P}_{det}$, κ has polynomial size and $V(body_Q) \not\subseteq rfacts_{\boldsymbol{P}}(\kappa)$ is testable in polynomial time. Therefore, it suffices to guess a valuation V and a node κ and verify that $V(body_Q) \not\subseteq rfacts_{\boldsymbol{P}}(\kappa)$.

Next, we show that $\mathbf{PC}^{bag}{}_{<_{\mathcal{N}}}(\mathrm{CQ}, \mathcal{P}_{fin})$ is coNP-hard. We use a simple reduction from the problem that asks whether a given graph is *not* 3-colorable. The latter is coNP-hard, since 3-colorability is well-known to be NP-complete. Now, let G be an arbitrary undirected graph with n edges. We notice that unconnected nodes in G do not affect colorability. W.l.o.g., we can thus assume that G has no such nodes and can be encoded by a binary relation E. More specifically, we choose to encode the edges in G in a directed fashion, that is, by including in E, for all edges $\{u, v\}$ in G, either (u, v) or (v, u), but not both.

We are now ready to construct a CQ Q and distribution policy $\boldsymbol{P} \in \mathcal{P}_{fin}$. For this, we denote by $\ell(.) : E \to [n]$ an arbitrarily chosen labeling that assigns to each edge in E a unique number from $[n]$. We later use this labeling to reason about colorings for the end-nodes of particular edges.

Let Q be the boolean CQ over $\mathcal{D} = \{E_i^{(2)} \mid i \in [n]\}$, with $body_Q = \{E_{\ell(e)}(x_v, x_u) \mid (v, u) = e \in E\}$. We define distribution policy $\boldsymbol{P}$ over network $\mathcal{N} = [n]$ and universe $U = \{r, g, b\}$, with $rfacts_{\boldsymbol{P}}(j) = \{\boldsymbol{f} \mid \boldsymbol{f} \in Facts(\{E_i\}, U), i \in [n], j \neq i\} \cup \{E_j(r, r), E_j(g, g), E_j(b, b)\}$, for every node $j \in \mathcal{N}$.

Clearly, the reduction is polynomial. It remains to show that G is *not* 3-colorable if and only if $\boldsymbol{Q}$ is bag-parallel-correct under $\boldsymbol{P}$ with ordered network $\mathcal{N}$.

(If). Suppose that G is 3-colorable. Let ρ be this coloring (say over colors $\{r, g, b\}$). Then there is a valuation V for Q that encodes ρ. More specifically, V is defined $V(x_u) = \rho(u)$. It is easy to see—by the construction of V—that for all combinations of variables x_u, x_v occuring together in some atom $E_i(x_u, x_v) \in body_Q$: $V(x_u) \neq V(x_v)$. It now follows directly from the construction of $\boldsymbol{P}$ that no node in $\mathcal{N}$ can support V, and from Lemma 5.3(1), that Q is not bag-parallel-correct under $\boldsymbol{P}$.

(Only-if). Let V be an arbitrary valuation for Q over U. We notice that V encodes a coloring for G. Indeed, let ρ be the mapping from nodes in G to colors in U, where $\rho(u) = V(x_u)$, for all nodes u in G. Since G is not 3-colorable, it must be that $\rho(u) = \rho(v)$ for some adjacent nodes u, v in G. Since $body_Q$ encodes G, there is an atom $E_i(x_u, x_v) \in body_Q$ (or $E_i(x_v, x_u) \in body_Q$), with $V(x_u) = V(x_v)$. It is now easy to see that $V(body_Q) \subseteq rfacts_{\boldsymbol{P}}(i)$, by construction of $\boldsymbol{P}$. It follows from Lemma 5.3(1), that Q is bag-parallel-correct under $\boldsymbol{P}$.

(2) We first argue that $\textbf{PC-Trans}^{bag}{}_{<_\mathcal{N}}(\mathrm{CQ}^{\neq}, \mathrm{CQ}^{\neq})$ is in Π_2^p. The required algorithm follows immediately from Lemma 5.3(2). Indeed, we just need to verify that for every valuation V' there is a valuation V such that $V'(body_{Q'}) \subseteq V(body_Q)$. The latter test can be performed in polynomial time and hence the result follows.

We next show that $\textbf{PC-Trans}^{bag}{}_{<_\mathcal{N}}(\mathrm{CQ}^{\neq}, \mathrm{CQ})$ is Π_2^p-hard. The reduction is from the quantified boolean satisfiability problem for the respective level of the hierarchy. That is, satisfiability for formulas of the form $\varphi = \forall \boldsymbol{x} \exists \boldsymbol{y} \psi(\boldsymbol{x}; \boldsymbol{y})$.

Let φ be such a formula.

(Encoding of ψ). For the construction of Q and Q', we first describe how ψ can be encoded as a set of atoms over schema $\mathcal{D} = \{\mathtt{Or}^{(3)}, \mathtt{And}^{(3)}, \mathtt{Neg}^{(2)}\}$. We denote this encoding by $Enc(\psi)$. An example of the construction is given in Figure 3. For the definition of this encoding, we need to associate to every variable x in ψ a unique variable in **var**. However, for convenience, we simply assume $x \in \textbf{var}$. We are now ready to define $Enc(\cdot)$ inductively as follows. For a single variable x, $Enc(x) = (\emptyset, x)$.

For propositional formulas ψ_1, ψ_2, with $Enc(\psi_1) = (A, x)$ and $Enc(\psi_2) = (B, y)$:

$Enc(\neg\psi_1) = (A \cup \{\mathtt{Neg}(x, z)\}, z)$;

$Enc((\psi_1)) = (A, x)$;

$Enc(\psi_1 \wedge \psi_2) = (A \cup B \cup \{\mathtt{And}(x, y, z)\}, z)$; and

$Enc(\psi_1 \vee \psi_2) = (A \cup B \cup \{\mathtt{Or}(x, y, z)\}, z)$.

In the above construction, we always choose for z a fresh variable that is not used in $Enc(\psi_1)$ nor $Enc(\psi_2)$.

We note that the above construction is non-deterministic, and that $Enc(\cdot)$ defines a set of encodings for ψ rather than just one. In the remainder of the proof, we assume that one such encoding is chosen (arbitrarily) and refer to it by $Enc(\psi) = (\mathtt{tempEncoding}_\psi, x_t)$.

(Construction of Q and Q'). Rather then defining Q and Q' directly over $\mathcal{D}$, we consider a schema with as many copies of the relation names in $\mathcal{D}$ as there are occurrences of the relation names in $\mathtt{tempEncoding}_\psi$. More formally, let n_X denote the number of atoms with relation name $X \in \mathcal{D}$ in $\mathtt{tempEncoding}_\psi$, and let ℓ_X denote a bijective labeling function from atoms with relation name X in $\mathtt{tempEncoding}_\psi$, to a unique index in $[n_X]$. We define Q and Q' over schema $\mathcal{D}' = \{\mathtt{Or}_i^{(3)} \mid i \in [n_{\mathtt{Or}}]\} \cup \{\mathtt{And}_i^{(3)} \mid i \in [n_{\mathtt{And}}]\} \cup \{\mathtt{Neg}_i^{(2)} \mid i \in [n_{\mathtt{Neg}}]\} \cup \{\mathtt{Bool}^{(2)}\} \cup \{P_x^{(2)} \mid x \in \boldsymbol{x}\}$.

For the construction, we use the following gadgets. Here, x_t and x_f denote special variables (recall that we used x_t also in $Enc(\psi) = (\mathtt{tmpEncoding}_\psi, x_t)$). Intuitively, x_t represents "true" and x_f represents "false". We use "_" to denote a fresh variable that is used exactly once in the query.

$$
\begin{aligned}
\texttt{encoding} =& \{X_i(\boldsymbol{z}) \mid X(\boldsymbol{z}) \in \texttt{tmpEncoding}_\psi, \text{ with } i = \ell_X(X(\boldsymbol{z}))\}. \\
\texttt{invalid} =& \{\texttt{Or}_i(\boldsymbol{t}) \mid \boldsymbol{t} \in \{(x_t, x_f, x_f), (x_f, x_t, x_f), (x_t, x_t, x_f), (x_f, x_f, x_t)\}, i \in [n_{\texttt{Or}}]\} \\
& \cup \{\texttt{And}_i(\boldsymbol{t}) \mid \boldsymbol{t} \in \{(x_t, x_f, x_t), (x_f, x_t, x_t), (x_f, x_f, x_t), (x_t, x_t, x_f)\}, i \in [n_{\texttt{And}}]\} \\
& \cup \{\texttt{Neg}_i(\boldsymbol{t}) \mid \boldsymbol{t} \in \{(x_t, x_t), (x_f, x_f)\}, i \in [n_{\texttt{Neg}}]\}. \\
\texttt{surplus} =& \{\texttt{Or}_i(\_,\_,\_), \texttt{Or}_i(\_,\_,\_), \texttt{Or}_i(\_,\_,\_) \mid i \in [n_{\texttt{Or}}]\} \\
& \cup \{\texttt{And}_i(\_,\_,\_), \texttt{And}_i(\_,\_,\_), \texttt{And}_i(\_,\_,\_) \mid i \in [n_{\texttt{And}}]\} \\
& \cup \{\texttt{Neg}_i(\_,\_) \mid i \in [n_{\texttt{Neg}}]\}. \\
\texttt{all} =& \{\texttt{Or}_i(\boldsymbol{z}) \mid i \in [n_{\texttt{Or}}], \boldsymbol{z} \in \{x_t, x_f\}^3\} \\
& \cup \{\texttt{And}_i(\boldsymbol{z}) \mid i \in [n_{\texttt{And}}], \boldsymbol{z} \in \{x_t, x_f\}^3\} \\
& \cup \{\texttt{Neg}_i(\boldsymbol{z}) \mid i \in [n_{\texttt{Neg}}], \boldsymbol{z} \in \{x_t, x_f\}^2]\}. \\
\texttt{chosenx} =& \{P_x(y_{x,t}, x_t), P_x(y_{x,f}, x_f), P_x(y_{x,u}, x_t), P_x(y_{x,u}, x_f) \mid x \in \boldsymbol{x}\}. \\
\texttt{fixedx} =& \{P_x(y_{x,1}, x), P_x(y_{x,2}, x), P_x(\_,\_), P_x(\_,\_) \mid x \in \boldsymbol{x}\}.
\end{aligned}
$$

Intuitively, atoms $\texttt{And}_i(x, y, z)$, $\texttt{Or}_i(x, y, z)$, and $\texttt{Neg}_i(x, y)$ represent propositional formulas $z = x \wedge y$, $z = x \vee y$, and $y = \neg x$, respectively. Therefore, interpreting x_t as “true” and x_f as “false”, `all` represents *all* possible truth assignments for these formulas, including those that do not satisfy the formula; `encoding` represents exactly one truth assignment for each formula, following the structure of ψ; `invalid` represents exactly the invalid truth assignments; and `surplus` allows to encode the remaining truth assignments (i.e, those that are not invalid, and not encoded by `encoding`). Gadgets `chosenx` and `fixedx` are inspired by a technique used in [20], and will be used to encode fixed partial assignments for $\boldsymbol{x}$.

We are now ready to define Q and Q', both with boolean head. We start with Q'.

$$
\begin{aligned}
body_{Q'} =& \{\texttt{Bool}(x_t, x_f)\} \cup \texttt{all} \\
& \texttt{chosenx}
\end{aligned}
$$

Query Q is defined as follows.

$$
\begin{aligned}
body_Q =& \{\texttt{Bool}(x_t, x_f)\} \cup \texttt{encoding} \cup \texttt{invalid} \cup \texttt{surplus} \cup \texttt{fixedx} \\
Diseq_{Q'} =& \{y_{x,1} \neq y_{x,2} \mid x \in \boldsymbol{x}\}.
\end{aligned}
$$

It is easy to see that the construction of Q and Q' is only polynomial in the size of φ. Next, we show that φ is satisfiable if and only if bag-parallel-correctness transfers from Q to Q'.

(If). Let β be an arbitrary truth assignment for $\boldsymbol{x}$. We need to show that β can be extended to a satisfying truth assignment for ψ. To this end, let V' be the valuation where $V'(x_t) = 1$, $V'(x_f) = 0$, and where for every $x \in \boldsymbol{x}$, we set $V'(y_{x,t}) = 1$ and $V'(y_{x,f}) = V'(y_{x,u}) = 0$, if $\beta(x) = 1$; and $V'(y_{x,f}) = 1$ and $V'(y_{x,t}) = V'(y_{x,u}) = 0$ otherwise. Intuitively, the latter guarantees that (†) $V'(body_{Q'}) \subseteq V(body_Q)$ can only satisfy if V agrees with β on the truth assignment for $x \in \boldsymbol{x}$. Particularly, this is because, for every $x \in \boldsymbol{x}$, four distinct P_x-facts are included in $V'(body_Q)$, $body_Q$ has only four P_x atoms, and $V(y_{x,1})$ must be distinct from $V(y_{x,2})$.

Now bag-parallel-correctness transfer from Q to Q' implies existence of valuation V for Q with $V'(body_{Q'}) \subseteq V(body_Q)$, and from (†) it follows that $V(x) = \beta(x)$ for all $x \in \boldsymbol{x}$. Further, we observe that $V'(\mathtt{Bool}(x_t, x_f)\}) \subseteq V(body_Q)$ can only satisfy if $V(x_t) = 1$ and $V(x_f) = 0$. By construction of $body_Q$ and $body_{Q'}$, $V'(\mathtt{all}) \subseteq V(body_Q)$ implies $V'(\mathtt{all}) = V(\mathtt{encoding} \cup \mathtt{invalid} \cup \mathtt{surplus})$, and more specifically, that $V'(\mathtt{all}) \setminus V(\mathtt{invalid}) = V(\mathtt{encoding} \cup \mathtt{surplus})$. Therefore, and since $V'(x_t) = 1$, V must encode a satisfying truth assignment for ψ that agrees on the choices of β for $x \in \boldsymbol{x}$. It is now straightforward that φ is indeed satisfiable.

(Only-If). Let V' be an arbitrary valuation for Q' over some universe U.

If $V'(x_t) = V(x_f)$, the result is straightforward. Indeed, we chose V, with $V(x_t) = V(x_f) = V'(x_t)$, and use the fresh variables in surplus to obtain $V'(\mathtt{all}) \subseteq V(\mathtt{surplus})$. For $V'(\mathtt{chosenx})$ we observe that, for every $x \in \boldsymbol{x}$, $V'(\mathtt{chosenx})$ contains at most 3 P_x-facts, all of the form $P_x(a_i, b)$, with fixed b. We can thus choose $V(x) = b$, $V(y_{x_1}) = a_1$. For remaining facts, we can freely map the atoms over anonymous variables.

Otherwise, if $V'(x_t) \neq V'(x_f)$, we interpret $V'(x_t)$ as 1 and $V'(x_f)$ as 0. For the construction of V, we first satisfy $V'(\mathtt{chosenx}) \subseteq V(\mathtt{fixedx})$. We do this as follows. Let $x \in \boldsymbol{x}$. If $V'(y_{x,t}) \neq V'(y_{x,u})$, we choose $V(y_{x,1}) = V(y_{x,t})$ and $V(y_{x,2}) = V'(y_{x,u})$, else, if $V'(y_{x,f}) \neq V'(y_{x,u})$, we choose $V(y_{x,1}) = V(y_{x,f})$ and $V(y_{x,2}) = V'(y_{x,u})$. Otherwise, we map $V(y_{x,1}) = V(y_{x,t})$ and map $V(y_{x,2})$ to an arbitrary distinct variable from U. It is now easy to see that V can be further extended so that $V'(\mathtt{chosenx}) \subseteq V(\mathtt{fixedx})$ by satisfying the other facts using the P_x atoms with anonymous variables.

Let now β be the partial truth assignment for ψ, where $\beta(x) = V(x)$ for all $x \in \boldsymbol{x}$. We notice that β is over $\{0, 1\}$ by assumption. Then, satisfiability of φ ensures existence of an extension β' of β that satisfies φ. We use β' to further construct V as follows: $V(x) = \beta'(x)$. Since β' is a satisfying truth assignment, it follows from the construction of encoding that V can be further extended so that $V(\mathtt{encoding})$ includes only facts encoding valid truth assignments. It is now easy to see, since $V(\mathtt{invalid})$ encodes all invalid assignments, we can obtain $V'(body_{Q'}) \subseteq V(body_Q)$, by tuning the anonymous variables in surplus. Since now $V'(body_{Q'}) \subseteq V(body_Q)$, the result follows.

(3) We first show the following lemma:

Lemma 5.5. *For a CQ Q, bag-parallel-correctness transfers from Q to Q' over ordered networks if and only if a mapping θ for Q over $adom(body_{Q'})$ exists such that $body_{Q'} \subseteq \theta(body_Q)$.*

Proof. It is easy to see that bag-parallel-correctness and Lemma 5.3(2) imply existence of θ: just take V' as the identity function. It remains to argue the other direction. That is, that existence of θ implies the conditions in Lemma 5.3(2). For this, we observe that if θ exists, then for every valuation W' for Q' over an U' there is mapping $\rho : U' \mapsto U'$, such that $\rho(x) = W'(x)$, if $x \in body_{Q'}$, and $\rho(x) = a$, for some arbitrary value $a \in U'$, otherwise. Then take the valuation W defined as $W(x) = \rho \circ \theta(x)$. It is now easy to see that $W'(body_{Q'}) \subseteq W(body_Q)$. □

That $\textbf{PC-Trans}^{bag}{}_{<_N}(\mathbf{CQ}, \mathbf{CQ}^{\neq})$ is in NP now follows directly from Lemma 5.5. Indeed, one can easily guess θ, and verify in polynomial time if $body_{Q'} \subseteq \theta(body_Q)$.

We next show that $\textbf{PC-Trans}^{bag}{}_{<_N}(\mathbf{CQ}, \mathbf{CQ})$ is NP-hard. We use a reduction from graph 3-colorability. Let G be an arbitrary undirected graph. We label every edge in G with a unique label from $[n]$, where n is the number of edges in G. We construct boolean queries Q and Q' over schema $\mathcal{D} = \{E_i^{(2)} \mid i \in [n]\}$. For Q', we have:

$$body_{Q'} = \{E_i(x, y) \mid i \in [n] \text{ and } x, y \in \{x_r, x_g, x_b\}\}.$$

For an example of the construction, consider the formula $\forall x \exists y \psi(x,y)$ with $\psi(x,y) = \neg\,[(x \vee \neg y) \wedge y]$. For this formula, we obtain $\texttt{tempEncoding}_\psi = (\{\texttt{Neg}(y,z_1), \texttt{Or}(x,z_1,x_2), \texttt{And}(y,z_2,z_3), \texttt{Neg}(z_3,x_t)\}, x_t)$ by applying the inductive construction in the following order: $\neg y$, $x \vee \neg y$, $(x \vee \neg y) \wedge y$, etc. Queries Q and Q' are defined over schema $\mathcal{D}'$, which for this example equals

$$\mathcal{D}' = \{\texttt{Neg}_1^{(2)}, \texttt{Or}_1^{(3)}, \texttt{And}_1^{(3)}, \texttt{Neg}_2^{(2)}, \texttt{Bool}^{(2)}, P_x^{(2)}, P_y^{(2)}\}.$$

Clearly, $n_{\texttt{And}}$ and $n_{\texttt{Or}}$ equal one, while $n_{\texttt{Neg}}$ equals two, which is the reason why two copies of the Neg relation symbol are installed in $\mathcal{D}'$. Now taking as bijective labeling function $\ell_{\texttt{Neg}(y,z_1)} = \ell_{\texttt{Or}(x,z_1,x_2)} = \ell_{\texttt{And}(y,z_2,z_3)} = 1$ and $\ell_{\texttt{Neg}(z_3,x_t)} = 2$, we obtain the following sets of atoms:

$$\begin{aligned}
\texttt{encoding} &= \{\texttt{Neg}_1(y,z_1), \texttt{Or}_1(x,z_1,x_2), \texttt{And}_1(y,z_2,z_3), \texttt{Neg}_2(z_3,x_t)\}\\
\texttt{invalid} &= \{\texttt{Or}_1(x_t,x_f,x_f), \texttt{Or}_1(x_f,x_t,x_f), \texttt{Or}_1(x_t,x_t,x_f), \texttt{Or}_1(x_f,x_f,x_t),\\
&\quad \texttt{And}_1(x_t,x_f,x_t), \texttt{And}_1(x_f,x_t,x_t), \texttt{And}_1(x_f,x_f,x_t), \texttt{And}_1(x_t,x_t,x_f),\\
&\quad \texttt{Neg}_1(x_t,x_t), \texttt{Neg}_1(x_f,x_f), \texttt{Neg}_2(x_t,x_t), \texttt{Neg}_2(x_f,x_f)\}\\
\texttt{surplus} &= \{\texttt{Or}_1(_,_,_), \texttt{Or}_1(_,_,_), \texttt{Or}_1(_,_,_), \texttt{And}_1(_,_,_), \texttt{And}_1(_,_,_), \texttt{And}_1(_,_,_),\\
&\quad \texttt{Neg}_1(_,_), \texttt{Neg}_2(_,_)\}\\
\texttt{all} &= \{\texttt{Or}_1(x_t,x_t,x_t), \texttt{Or}_1(x_t,x_f,x_t), \texttt{Or}_1(x_f,x_t,x_t), \texttt{Or}_1(x_f,x_f,x_f),\\
&\quad \texttt{And}_1(x_t,x_t,x_t), \texttt{And}_1(x_t,x_f,x_f), \texttt{And}_1(x_f,x_t,x_f), \texttt{And}_1(x_f,x_f,x_f),\\
&\quad \texttt{Neg}_1(x_t,x_f), \texttt{Neg}_1(x_f,x_t), \texttt{Neg}_2(x_t,x_f), \texttt{Neg}_2(x_f,x_t)\} \cup \texttt{invalid}\\
\texttt{chosenx} &= \{P_x(y_{x,t},x_t), P_x(y_{x,f},x_f), P_x(y_{x,u},x_t), P_x(y_{x,u},x_f)\}\\
\texttt{fixedx} &= \{P_x(y_{x,1},x), P_x(y_{x,2},x), P_x(_,_), P_x(_,_)\}
\end{aligned}$$

Finally, based on the above gadgets, we obtain the following queries Q and Q'.

$$\begin{aligned}
Q = () \leftarrow &\texttt{Bool}(x_t,x_f), \texttt{Neg}_1(y,z_1), \texttt{Or}_1(x,z_1,x_2), \texttt{And}_1(y,z_2,z_3), \texttt{Neg}_2(z_3,x_t),\\
&\texttt{Or}_1(x_t,x_f,x_f), \texttt{Or}_1(x_f,x_t,x_f), \texttt{Or}_1(x_t,x_t,x_f), \texttt{Or}_1(x_f,x_f,x_t),\\
&\texttt{And}_1(x_t,x_f,x_t), \texttt{And}_1(x_f,x_t,x_t), \texttt{And}_1(x_f,x_f,x_t), \texttt{And}_1(x_t,x_t,x_f),\\
&\texttt{Neg}_1(x_t,x_t), \texttt{Neg}_1(x_f,x_f), \texttt{Neg}_2(x_t,x_t), \texttt{Neg}_2(x_f,x_f), \texttt{Or}_1(_,_,_),\\
&\texttt{Or}_1(_,_,_), \texttt{Or}_1(_,_,_), \texttt{And}_1(_,_,_), \texttt{And}_1(_,_,_), \texttt{And}_1(_,_,_), \texttt{Neg}_1(_,_),\\
&\texttt{Neg}_2(_,_), P_x(y_{x,1},x), P_x(y_{x,2},x), P_x(_,_), P_x(_,_), y_{x,1} \neq y_{x,2}.\\
Q' = () \leftarrow &\texttt{Bool}(x_{,t}, x_f),\\
&\texttt{Or}_1(x_t,x_t,x_t), \texttt{Or}_1(x_t,x_f,x_t), \texttt{Or}_1(x_f,x_t,x_t), \texttt{Or}_1(x_f,x_f,x_f),\\
&\texttt{And}_1(x_t,x_t,x_t), \texttt{And}_1(x_t,x_f,x_f), \texttt{And}_1(x_f,x_t,x_f), \texttt{And}_1(x_f,x_f,x_f),\\
&\texttt{Neg}_1(x_t,x_f), \texttt{Neg}_1(x_f,x_t), \texttt{Neg}_2(x_t,x_f), \texttt{Neg}_2(x_f,x_t),\\
&\texttt{Or}_1(x_t,x_f,x_f), \texttt{Or}_1(x_f,x_t,x_f), \texttt{Or}_1(x_t,x_t,x_f), \texttt{Or}_1(x_f,x_f,x_t),\\
&\texttt{And}_1(x_t,x_f,x_t), \texttt{And}_1(x_f,x_t,x_t), \texttt{And}_1(x_f,x_f,x_t), \texttt{And}_1(x_t,x_t,x_f),\\
&\texttt{Neg}_1(x_t,x_t), \texttt{Neg}_1(x_f,x_f), \texttt{Neg}_2(x_t,x_t), \texttt{Neg}_2(x_f,x_f),\\
&P_x(y_{x,t},x_t), P_x(y_{x,f},x_f), P_x(y_{x,u},x_t), P_x(y_{x,u},x_f).
\end{aligned}$$

Fig. 3. Application of the construction of queries Q and Q' as described in the proof of Theorem 5.4(2) on formula $\forall x \exists y \neg\,[(x \vee \neg y) \wedge y]$.

Intuitively, Q' encodes for each edge in G all possible red, green, blue colorings (not necessarily only valid ones). Notice that for every edge, there are 9 possible colorings. For query Q, 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 are now ready to define query Q:

$$\mathtt{body}_Q = \{E_i(x_u, x_v) \mid E(u,v) \in G \text{ having label } i\} \cup \mathtt{invalidE} \cup \mathtt{surplusE}.$$

For every edge $E(u,v)$ in G (say with label i) there are 9 atoms in Q each corresponding to one specific coloring of $E(u,v)$:

- the atom $E_i(x_u, x_v)$ stemming from the edge $E(u,v) \in G$; this atom corresponds to the chosen 3-coloring;
- the atoms in `invalidE` corresponding to invalid colorings; and,
- the atoms in `surplusE` corresponding to surplus colorings, that is, valid colorings that will not be used.

Intuitively, $body_{Q'} \subseteq \theta(body_Q)$ 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.

The reduction is clearly polynomial in the size of G. Next, we show that G is 3-colorable if and only if $Q \xrightarrow{\text{bag}} Q'$.

(If). Suppose $Q \xrightarrow{\text{bag}} Q'$. By Lemma 5.5, there is a θ such that $body_{Q'} \subseteq \theta(body_Q)$. Now define the mapping ρ as $\rho(u) = \theta(x_u)$ for every node u in G (using x_r, x_b, x_g as colorings). We argue that ρ is a 3-coloring of G. Towards a contradiction, assume $\rho(u) = \rho(v)$ for some edge (u,v) in G with label i. This would imply that for atom E_i there are at most 8 colorings present in $\theta(body_Q)$ whereas there are 9 colorings in $\theta(body_{Q'})$. So, $body_{Q'} \not\subseteq \theta(body_Q)$ which leads to the desired contradiction.

(Only-if). Assume G is 3-colorable and let ρ be a valid coloring for G over colors $\{r, g, b\}$. By Lemma 5.5, it suffices to show that there is a mapping θ for which $body_{Q'} \subseteq \theta(body_Q)$. To this end, define θ as follows: $\theta(x_u) = x_{\rho(u)}$ for every node u in G. Then assign values to all other variables to encompass the 8 additional colorings of every edge. By construction of Q there are 8 additional atoms for every edge and it is therefore possible to do so while ensuring that $body_{Q'} \subseteq \theta(body_Q)$. ◻

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 semantic 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 and an NP-hard lower bound. 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 obtain a strict 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.

ACKNOWLEDGMENTS

This work is partly funded by FWO-grant G062721N and G055219N.

REFERENCES

[1] Foto N. Afrati, Manas R. Joglekar, Christopher Ré, Semih Salihoglu, and Jeffrey D. Ullman. 2017. GYM: A Multiround Distributed Join Algorithm. In *ICDT*. 4:1–4:18.
[2] Foto N. Afrati, Anish Das Sarma, Semih Salihoglu, and Jeffrey D. Ullman. 2013. Upper and Lower Bounds on the Cost of a Map-Reduce Computation. *PVLDB* 6, 4 (2013), 277–288.
[3] Foto N. Afrati and Jeffrey D. Ullman. 2010. Optimizing joins in a map-reduce environment. In *EDBT*. 99–110.
[4] Tom J. Ameloot, Gaetano Geck, Bas Ketsman, Frank Neven, and Thomas Schwentick. 2016. Data partitioning for single-round multi-join evaluation in massively parallel systems. *SIGMOD Record* 45, 1 (2016), 33–40.
[5] Tom J. Ameloot, Gaetano Geck, Bas Ketsman, Frank Neven, and Thomas Schwentick. 2017. Parallel-Correctness and Transferability for Conjunctive Queries. *J. ACM* 64, 5, 36:1–36:38.
[6] Paul Beame, Paraschos Koutris, and Dan Suciu. 2014. Skew in Parallel Query Processing. In *PODS*. ACM, 212–223.
[7] Surajit Chaudhuri and Moshe Y. Vardi. 1993. Optimization of *Real* Conjunctive Queries. In *PODS*. ACM Press, 59–70.
[8] Sara Cohen. 2009. Equivalence of queries that are sensitive to multiplicities. *VLDB J.* 18, 3 (2009), 765–785.
[9] Sumit Ganguly, Abraham Silberschatz, and Shalom Tsur. 1990. A Framework for the Parallel Processing of Datalog Queries. In *SIGMOD*. ACM Press, 143–152.
[10] Gaetano Geck, Bas Ketsman, Frank Neven, and Thomas Schwentick. 2016. Parallel-Correctness and Containment for Conjunctive Queries with Union and Negation. In *ICDT*. 9:1–9:17.
[11] Gaetano Geck, Frank Neven, and Thomas Schwentick. 2020. Distribution Constraints: The Chase for Distributed Data. In *23rd International Conference on Database Theory (ICDT) (LIPIcs, Vol. 155)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 13:1–13:19.
[12] Hadoop. 2024. (2024). https://hadoop.apache.org/.
[13] Bas Ketsman, Aws Albarghouthi, and Paraschos Koutris. 2020. Distribution Policies for Datalog. *Theory Comput. Syst.* 64, 5 (2020), 965–998.
[14] Bas Ketsman, Frank Neven, and Brecht Vandevoort. 2018. Parallel-Correctness and Transferability for Conjunctive Queries under Bag Semantics. In *21st International Conference on Database Theory, (ICDT) (LIPIcs, Vol. 98)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 18:1–18:16.
[15] Bas Ketsman and Dan Suciu. 2017. A Worst-Case Optimal Multi-Round Algorithm for Parallel Computation of Conjunctive Queries. In *PODS*. 417–428.
[16] Paraschos Koutris and Dan Suciu. 2011. Parallel evaluation of conjunctive queries. In *PODS*. 223–234.
[17] Rimma Nehme and Nicolas Bruno. 2011. Automated Partitioning Design in Parallel Database Systems. In *SIGMOD*. ACM, 1137–1148.
[18] Jun Rao, Chun Zhang, Nimrod Megiddo, and Guy Lohman. 2002. Automating Physical Database Design in a Parallel Database. In *SIGMOD*. 558–569.
[19] Bruhathi Sundarmurthy, Paraschos Koutris, and Jeffrey F. Naughton. 2021. Locality-Aware Distribution Schemes. In *24th International Conference on Database Theory, (ICDT) (LIPIcs, Vol. 186)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 22:1–22:25.
[20] Ron van der Meyden. 1992. The Complexity of Querying Indefinite Data about Linearly Ordered Domains. In *PODS*. 331–345.
[21] Reynold S. Xin, Josh Rosen, Matei Zaharia, Michael J. Franklin, Scott Shenker, and Ion Stoica. 2013. Shark: SQL and rich analytics at scale. In *SIGMOD*. 13–24.