Parallel-Correctness and Transferability for Conjunctive Queries

Tom J. Ameloot[*]
Hasselt University & transnational University of Limburg
tom.ameloot@uhasselt.be

Gaetano Geck
TU Dortmund University
gaetano.geck@tu-dortmund.de

Bas Ketsman[†]
Hasselt University & transnational University of Limburg
bas.ketsman@uhasselt.be

Frank Neven
Hasselt University & transnational University of Limburg
frank.neven@uhasselt.be

Thomas Schwentick
TU Dortmund University
thomas.schwentick@udo.edu

ABSTRACT

A dominant cost for query evaluation in modern massively distributed systems is the number of communication rounds. For this reason, there is a growing interest in single-round multiway join algorithms where data is first reshuffled over many servers and then evaluated in a parallel but communication-free way. The reshuffling itself is specified as a distribution policy. We introduce a correctness condition, called *parallel-correctness*, for the evaluation of queries w.r.t. a distribution policy. We study the complexity of parallel-correctness for conjunctive queries as well as transferability of parallel-correctness between queries. We also investigate the complexity of transferability for certain families of distribution policies, including, for instance, the Hypercube distribution.

Categories and Subject Descriptors

H.2 [**Database Management**]: Languages; H.2 [**Database Management**]: Systems—*Distributed databases*

Keywords

Distributed databases; Parallel query evaluation; One-round evaluation; Distribution policies

1. INTRODUCTION

In traditional database systems, the complexity of query processing for large datasets is mainly determined by the number of IO requests to external memory. A factor dominating complexity in modern massively distributed database systems, however, is the number of communication steps [6]. Motivated by recent in-memory systems like Spark [12] and Shark [14], Koutris and Suciu introduced the massively parallel communication model (MPC) [11] where computation proceeds in a sequence of parallel steps each followed by global synchronization of all servers. In this model, evaluation of conjunctive queries [5, 11] and skyline queries [2] has been considered.

Of particular interest in the MPC model are the queries that can be evaluated in one round of communication. Recently, Beame, Koutris and Suciu [6] proved a matching upper and lower bound for the amount of communication needed to compute a full conjunctive query without self-joins in one communication round. The upper bound is provided by a randomized algorithm called Hypercube which uses a technique that can be traced back to Ganguli, Silberschatz, and Tsur [9] and is described in the context of map-reduce by Afrati and Ullman [3]. The Hypercube algorithm evaluates a conjunctive query $\mathcal{Q}$ by first reshuffling the data over many servers and then evaluating $\mathcal{Q}$ at each server in a parallel but communication-free manner. The reshuffling is specified by a distribution policy (hereafter, called Hypercube distribution) and is based on the structure of $\mathcal{Q}$. In particular, the Hypercube distribution partitions the space of all complete valuations of $\mathcal{Q}$ over the computing servers in an instance independent way through hashing of domain values. A property of Hypercube distributions is that for any instance I, the central execution of $\mathcal{Q}(I)$ always equals the union of the evaluations of $\mathcal{Q}$ at every computing node (or server).[1]

In this paper, we introduce a general framework for reasoning about one-round evaluation algorithms under *arbitrary* distribution policies.[2] Distribution policies (formally

[*]Postdoctoral Fellow of the Research Foundation - Flanders (FWO).

[†]PhD Fellow of the Research Foundation - Flanders (FWO).

PODS'15, May 31–June 4, 2015, Melbourne, Victoria, Australia.

ACM 978-1-4503-2757-2/15/05 ...$15.00.
http://dx.doi.org/10.1145/2745754.2745759.

[1]We emphasize that, for a query $\mathcal{Q}$, there is no single Hypercube distribution but rather a family of distributions as the concrete instantiation depends on choices regarding the address space of servers.

[2]Our aim is to study one-round evaluation algorithms, not to *advocate* them. We plan further investigation that also takes multi-round algorithms into account. Furthermore,

defined in Section 2) are functions mapping input facts to sets of nodes (servers) in the network. We introduce the following correctness property for queries and distribution policies: a query $\mathcal{Q}$ is *parallel-correct* for a given distribution policy $\boldsymbol{P}$, when for any instance I, the evaluation of $\mathcal{Q}(I)$ equals the union of the evaluation of $\mathcal{Q}$ over the distribution of I under policy $\boldsymbol{P}$. We focus on conjunctive queries and study the complexity of deciding parallel-correctness. We show that the latter problem is equivalent to testing whether the facts in every *minimal* valuation of the conjunctive query are mapped to a same node in the network by the distribution policy. For various representations of distributions policies, we then show that testing parallel-correctness is in Π_2^P. We provide a matching lower bound via a reduction from the Π_2^P-complete Π_2-QBF-problem.

One-round evaluation algorithms, like Hypercube, redistribute data for the evaluation of every query. For scenarios where queries are executed in sequence, it makes sense to study cases where the same data distribution can be used to evaluate multiple queries. We formalize this as *parallel-correctness transfer* between queries. In particular, parallel-correctness *transfers* from $\mathcal{Q}$ to $\mathcal{Q}'$ when $\mathcal{Q}'$ is parallel-correct under every distribution policy for which $\mathcal{Q}$ is parallel-correct. We characterize transferability for conjunctive queries by a (value-based) containment condition for minimal valuations of $\mathcal{Q}'$ and $\mathcal{Q}$, and use this characterization to obtain a Π_3^P upper bound for transferability. Again, we obtain a matching lower bound, this time via a reduction from the Π_3^P-complete Π_3-QBF-problem. We obtain a (presumably) better complexity, NP-completeness, in the case that $\mathcal{Q}$ is *strongly minimal*, i.e., when all its valuations are minimal. Examples of strongly minimal CQs include the full conjunctive queries and those without self-joins. At the heart of the upper bound proof lies the insight that the above mentioned value-based inclusion w.r.t. minimal valuations reduces to a syntactic inclusion of $\mathcal{Q}'$ in $\mathcal{Q}$ modulo a variable renaming when $\mathcal{Q}$ is strongly minimal. We obtain that deciding strong minimality is NP-complete as well.

Finally, we study parallel-correctness transfer from $\mathcal{Q}$ to $\mathcal{Q}'$ w.r.t. a specific family $\mathcal{F}$ of distribution policies rather than the set of *all* distribution policies. We show that it is NP-complete to decide whether $\mathcal{Q}'$ is parallel-correct for a given family $\mathcal{F}$ if this family has the following two properties: it is $\mathcal{Q}$-generous (for each, not only for minimal, valuation of $\mathcal{Q}$, its facts occur at some node) and $\mathcal{Q}$-scattered (for every instance some distribution has, at every node, only facts from one valuation). It is easy to see that the family of Hypercube distributions for a given CQ $\mathcal{Q}$ satisfies these properties, which implies that deciding transferability for Hypercube distributions is NP-complete, as well.

We complete our framework by sketching a declarative specification formalism for distribution policies, illustrated with the specification of Hypercube distributions.

Outline. We introduce the necessary definitions in Section 2. We study parallel-correctness in Section 3 and transferability in Section 4. We examine families of distribution policies including the Hypercube distribution in Section 5. We conclude in Section 6. We give at least sketches of most proofs but defer some proof details to the full version of this paper.

our emphasis is on *reasoning* about queries and distribution policies, not on the development of good distribution policies.

2. DEFINITIONS

Queries and instances.

We assume an infinite set **dom** of data values that can be represented by strings over some fixed alphabet. A *database schema* $\mathcal{D}$ is a finite set of relation names R where every R has arity $ar(R)$. We call $R(\mathbf{t})$ a *fact* when R is a relation name and $\mathbf{t}$ a tuple in **dom**. We say that a fact $R(d_1, \ldots, d_k)$ is *over* a database schema $\mathcal{D}$ if $R \in \mathcal{D}$ and $ar(R) = k$. By *facts*$(\mathcal{D})$, we denote the set of possible facts over schema $\mathcal{D}$. A *(database) instance* I over $\mathcal{D}$ is a finite set of facts over $\mathcal{D}$. By $adom(I)$ we denote the set of data values occurring in I. A *query $\mathcal{Q}$ over input schema $\mathcal{D}_1$ and output schema* $\mathcal{D}_2$ is a generic mapping from instances over $\mathcal{D}_1$ to instances over $\mathcal{D}_2$. Genericity means that for every permutation π of **dom** and every instance I, $\mathcal{Q}(\pi(I)) = \pi(\mathcal{Q}(I))$.

Conjunctive queries.

Let **var** be the universe of variables, disjoint from **dom**. An *atom* is of the form $R(\mathbf{x})$, where R is a relation name and $\mathbf{x}$ is a tuple of variables in **var**. We say that $R(x_1, \ldots, x_k)$ is an atom *over* schema $\mathcal{D}$ if $R \in \mathcal{D}$ and $k = ar(R)$.

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

$$T(\mathbf{x}) \leftarrow R_1(\mathbf{y}_1), \ldots, R_n(\mathbf{y}_n),$$

where every $R_i(\mathbf{y}_i)$ is an atom over $\mathcal{D}$, and $T(\mathbf{x})$ is an atom for which $T \notin \mathcal{D}$. Additionally, for safety, we require that every variable in $\mathbf{x}$ occurs in some $\mathbf{y}_i$. We refer to the *head atom* $T(\mathbf{x})$ by $head_{\mathcal{Q}}$, and denote the set of body atoms $R_i(\mathbf{y}_i)$ by $body_{\mathcal{Q}}$.

A conjunctive query is called *full* if all variables of the body also occur in the head. We say that a CQ is *without self-joins* when all of its atoms have a distinct relation name.

We denote by $vars(\mathcal{Q})$ the set of all variables occurring in $\mathcal{Q}$. A *valuation* for a conjunctive query is a total function $V : vars(\mathcal{Q}) \to \mathbf{dom}$ that maps each variable of $\mathcal{Q}$ to a data value. We say that V *requires* or *needs* the facts $V(body_{\mathcal{Q}})$ for $\mathcal{Q}$. A valuation V is said to be *satisfying* for $\mathcal{Q}$ on instance I, when all the facts required by V for $\mathcal{Q}$ are in I. In that case, V *derives* the fact $V(head_{\mathcal{Q}})$. The result of $\mathcal{Q}$ on instance I, denoted $\mathcal{Q}(I)$, is defined as the set of facts that can be derived by satisfying valuations for $\mathcal{Q}$ on I. We note that, as we do not allow negation, all conjunctive queries are monotone.

We frequently compare different valuations for a query $\mathcal{Q}$ with respect to their required sets of facts. For two valuations V_1, V_2 for a CQ $\mathcal{Q}$, we write $V_1 \leq_Q V_2$ if $V_1(head_{\mathcal{Q}}) = V_2(head_{\mathcal{Q}})$ and $V_1(body_{\mathcal{Q}}) \subseteq V_2(body_{\mathcal{Q}})$. We write $V_1 <_Q V_2$ if furthermore $V_1(body_{\mathcal{Q}}) \subsetneq V_2(body_{\mathcal{Q}})$ holds.

A *substitution* is a mapping from variables to variables, which is generalized to tuples, atoms and conjunctive queries in the natural fashion [1].[3] We denote the composition of functions in the usual way, i.e., $(f \circ g)(x) \stackrel{\text{def}}{=} f(g(x))$.

The following notion is fundamental for the development in the rest of the paper:

[3]As we only consider CQs without constants, substitutions do not map variables to constants.

Definition 1. A *simplification* of a conjunctive query $\mathcal{Q}$ is a substitution $\theta : vars(\mathcal{Q}) \to vars(\mathcal{Q})$ for which $head_{\theta(\mathcal{Q})} = head_{\mathcal{Q}}$ and $body_{\theta(\mathcal{Q})} \subseteq body_{\mathcal{Q}}$.

A simplification is thus a homomorphism from $\mathcal{Q}$ to $\mathcal{Q}$ and by the homomorphism theorem [1] (and the trivial embedding from $\theta(\mathcal{Q})$ to $\mathcal{Q}$), $\mathcal{Q}$ and $\theta(\mathcal{Q})$ are equivalent. Of course, the identity substitution is always a simplification.

EXAMPLE 2.1. *We give a few examples to illustrate simplifications. Consider the query*

$$T(x) \leftarrow R(x,x), R(x,y), R(x,z).$$

Then $\theta_1 = \{x \mapsto x, y \mapsto y, z \mapsto y\}$ *as well as* $\theta_2 = \{x \mapsto x, y \mapsto x, z \mapsto x\}$ *are simplifications. For the query*

$$T(x) \leftarrow R(x,y), R(y,y), R(z,z), R(u,u),$$

possible simplifications are $\theta_3 = \{x \mapsto x, y \mapsto y, z \mapsto y, u \mapsto z\}$ *and* $\theta_4 = \{x \mapsto x, y \mapsto y, z \mapsto y, u \mapsto y\}$*. For the query* $T(x) \leftarrow R(x,y), R(y,z)$ *there are no simplifications besides the identity.* $\square$

The notion of simplification is closely related to foldings as defined by Chandra and Merlin [7]. In particular, a *folding* of a conjunctive query $\mathcal{Q}$ is a simplification θ that is idempotent. That is, $\theta^2 = \theta$. Intuitively, the idempotence means that when θ gives a new name to a variable then it sticks to it. Notice that in Example 2.1 simplifications θ_1, θ_2, θ_4 are foldings but θ_3 is not as $\theta_3(u) = z \neq y = \theta_3(\theta_3(u))$.

Networks, data distribution, and policies.

A *network* $\mathcal{N}$ is a nonempty finite set of values from **dom**, which we call *(computing) nodes*.

A *distribution policy* $\boldsymbol{P}$ for a database schema $\mathcal{D}$ and a network $\mathcal{N}$ is a total function mapping facts from $facts(\mathcal{D})$ to sets of nodes.[4] For an instance I over $\mathcal{D}$, let $dist_{\boldsymbol{P},I}$ denote the function that maps each $\kappa \in \mathcal{N}$ to $\{\boldsymbol{f} \in I \mid \kappa \in \boldsymbol{P}(\boldsymbol{f})\}$, that is, the set of facts assigned to it by $\boldsymbol{P}$. We sometimes refer to $dist_{\boldsymbol{P},I}(\kappa)$ as a *data chunk*.

In this paper, we do not always explicitly give names to schemas and networks but tacitly assume they are understood from the queries and the distribution policies under consideration, respectively.

We do *not* always expect that distribution policies $\boldsymbol{P}$ are given as part of the input by exhaustive enumeration of all pairs $(\kappa, \boldsymbol{f})$, for which $\kappa \in \boldsymbol{P}(\boldsymbol{f})$. We also consider mechanisms, where instead the distribution policy is implicitly represented by a given "black box" procedure. While there are many possible ways to represent distribution policies, either as functions or as relations belonging to various complexity classes, in this paper, we only consider one such class. In particular, we define the class $\mathcal{P}_{\text{nrel}}$ where each distribution $\boldsymbol{P}$ is represented by a NP-testable relation, that on input $(\kappa, \boldsymbol{f})$ yields "true" if and only if $\kappa \in \boldsymbol{P}(\boldsymbol{f})$. We will discuss declarative ways to specify distribution policies in a non-black-box fashion in Section 5.

The definition of a distribution policy is borrowed from Ameloot et al. [4] (but already surfaces in the work of Zinn et al. [15]), where distribution policies are used to define the class of policy-aware transducer networks.

[4]Notice that our formalization allows to 'skip' facts by mapping them to the empty set of nodes. This is, for instance, the case for a Hypercube distribution (cf. Section 5), which skips facts that are not essential to evaluate the query at hand.

3. PARALLEL-CORRECTNESS

In this section, we introduce and study the notion of parallel-correctness, which is central to this paper.

Definition 2. A query $\mathcal{Q}$ is *parallel-correct on instance I under distribution policy $\boldsymbol{P}$*, if $\mathcal{Q}(I) = \bigcup_{\kappa \in \mathcal{N}} \mathcal{Q}(dist_{\boldsymbol{P},I}(\kappa))$.

That is, the centralized execution of $\mathcal{Q}$ on I is the same as taking the union of the results obtained by executing $\mathcal{Q}$ at every computing node. Next, we lift parallel-correctness to all instances.

Definition 3. A query $\mathcal{Q}$ is *parallel-correct under distribution policy $\boldsymbol{P}$*, if $\mathcal{Q}$ is parallel-correct on all input instances under $\boldsymbol{P}$.

Of course, when a query $\mathcal{Q}$ is parallel-correct under $\boldsymbol{P}$, there is a direct one-round evaluation algorithm for every instance. Indeed, the algorithm first distributes (reshuffles) the data over the computing nodes according to $\boldsymbol{P}$ and then evaluates Q in a subsequent parallel step at every computing node. Notice that as $\boldsymbol{P}$ is defined on the granularity of a fact, the reshuffling does not depend on the current distribution of the data and can be done in parallel as well.

While Definitions 2 and 3 are in terms of general queries, in the rest of this section, we only consider conjunctive queries. It is easy to see that a CQ $\mathcal{Q}$ is parallel-correct under distribution policy $\boldsymbol{P}$ if for each valuation for $\mathcal{Q}$ the required facts meet at some node, i.e., if the following condition holds:

(C0) for every valuation V for $\mathcal{Q}$,

$$\bigcap_{\boldsymbol{f} \in V(body_{\mathcal{Q}})} \boldsymbol{P}(\boldsymbol{f}) \neq \emptyset.$$

Even though (C0) is sufficient for parallel-correctness, it is not necessary (c.f., Example 3.2). It turns out that for a semantical characterization only valuations have to be considered that are minimal in the following sense.

Definition 4. Let $\mathcal{Q}$ be a CQ. A valuation V for $\mathcal{Q}$ is *minimal* for $\mathcal{Q}$ if there exists no valuation V' for $\mathcal{Q}$ such that $V' <_{\mathcal{Q}} V$.

The next lemma now states the targeted characterization:

LEMMA 3.1. *A CQ $\mathcal{Q}$ is parallel-correct under distribution policy $\boldsymbol{P}$ if and only if the following holds:*

(C1) for every minimal valuation V for $\mathcal{Q}$,

$$\bigcap_{\boldsymbol{f} \in V(body_{\mathcal{Q}})} \boldsymbol{P}(\boldsymbol{f}) \neq \emptyset.$$

PROOF (SKETCH). (if) Assume (C1) holds. Because of monotonicity, we only need to show that, for every instance I, $\mathcal{Q}(I) \subseteq \bigcup_{\kappa \in \mathcal{N}} \mathcal{Q}(dist_{\boldsymbol{P},I}(\kappa))$. To this end, let $\boldsymbol{f}$ be a fact that is derived by some valuation V for $\mathcal{Q}$ over I. Then, there is also a minimal valuation V' that is satisfying on I and which derives $\boldsymbol{f}$. Because of condition (C1), there is a node κ where all facts required for V' meet. Hence, $\boldsymbol{f} \in \bigcup_{\kappa \in \mathcal{N}} \mathcal{Q}(dist_{\boldsymbol{P},I}(\kappa))$.

(only-if) Proof by contraposition. Suppose that there is a minimal valuation V' for $\mathcal{Q}$ for which the required facts do not meet under $\boldsymbol{P}$. Consider $V'(body_{\mathcal{Q}})$ as input instance. Then, by definition of minimality, there is no valuation that agrees on the head-variables and is satisfied on one of the chunks of $V'(body_{\mathcal{Q}})$ under $\boldsymbol{P}$. So, $\mathcal{Q}$ is not parallel-correct. $\square$

EXAMPLE 3.2. *For a simple example of a minimal valuation and a non-minimal valuation, consider the CQ $\mathcal{Q}$,*

$$T(x,z) \leftarrow R(x,y), R(y,z), R(x,x).$$

Both $V = \{x \mapsto a, y \mapsto b, z \mapsto a\}$ and $V' = \{x \mapsto a, y \mapsto a, z \mapsto a\}$ are valuations for $\mathcal{Q}$. Notice that both valuations agree on the head-variables of $\mathcal{Q}$, but they require different sets of facts. In particular, for V to be satisfying on I, instance I must contain the facts $R(a,b), R(b,a)$, and $R(a,a)$, while V' only requires I to contain $R(a,a)$. This observation implies that V is not minimal for $\mathcal{Q}$. Further, as V' requires only one fact for $\mathcal{Q}$, V' must be minimal for $\mathcal{Q}$.

We next argue that (C0) is not a necessary condition for parallel-correctness. Indeed, take $\mathcal{N} = \{1,2\}$ and $\boldsymbol{P}$ as the distribution policy mapping every fact except $R(a,b)$ onto node 1 and every fact except $R(b,a)$ onto node 2. Consider the valuations V and $W = \{x \mapsto b, y \mapsto a, z \mapsto b\}$. Then, $R(a,b)$ and $R(b,a)$ do not meet under $\boldsymbol{P}$, thus violating condition (C0). It remains to argue that $\mathcal{Q}$ is parallel-correct under $\boldsymbol{P}$. For every minimal valuation U, either $\bigcap_{\boldsymbol{f} \in U(body_{\mathcal{Q}})} \boldsymbol{P}(\boldsymbol{f}) \neq \emptyset$ or U requires both $R(a,b)$ and $R(b,a)$. In the latter case U is either valuation V or W as defined above, which are not minimal. Thus, by Lemma 3.1, query $\mathcal{Q}$ is parallel-correct under $\boldsymbol{P}$. $\square$

Unfortunately, condition (C1) is complexity-wise more involved than (C0) as minimality of V needs to be tested. The lower bound in Theorem 3.6 below indicates that this can, in a sense, not be avoided.

Towards an upper bound for the complexity of parallel-correctness, we first discuss how minimality of a valuation can be tested. Obviously, this notion is related to the (classical) notion of minimality for conjunctive queries, as we will make precise next. First, recall that a CQ $\mathcal{Q}$ is *minimal* if there is no equivalent CQ with strictly less atoms.

LEMMA 3.3. *Let $\mathcal{Q}$ be a conjunctive query. For every injective valuation V for $\mathcal{Q}$, it holds that V is minimal if and only if $\mathcal{Q}$ is minimal.*

PROOF. In the following let $\mathcal{Q}$ be a CQ. We show that there is a non-minimal injective valuation V for $\mathcal{Q}$ if and only if $\mathcal{Q}$ is not minimal.

(if) Suppose that $\mathcal{Q}$ is not minimal. Then, by [7] there is a folding h for $\mathcal{Q}$, where $body_{h(\mathcal{Q})} \subsetneq body_{\mathcal{Q}}$ and $head_{h(\mathcal{Q})} = head_{\mathcal{Q}}$. Let V be an arbitrary injective valuation for $\mathcal{Q}$. Injectivity implies that $|V(body_{\mathcal{Q}})| = |body_{\mathcal{Q}}|$, that is the number of facts in $V(body_{\mathcal{Q}})$ equals the number of atoms in $body_{\mathcal{Q}}$.

Since $h(\mathcal{Q})$ only has variables that also appear in $\mathcal{Q}$, V is a valuation for $h(\mathcal{Q})$ as well. However, thanks to $body_{h(\mathcal{Q})} \subsetneq body_{\mathcal{Q}}$, $h(body_{\mathcal{Q}})$ has fewer atoms than $body_{\mathcal{Q}}$, therefore $(V \circ h)(body_{\mathcal{Q}})$ has fewer facts than $V(body_{\mathcal{Q}})$. Thus, $(V \circ h)$ is a counterexample for the minimality of V, since $(V \circ h)(body_{\mathcal{Q}}) = V(body_{h(\mathcal{Q})}) \subseteq V(body_{\mathcal{Q}})$ and $(V \circ h)(head_{\mathcal{Q}}) = V(head_{h(\mathcal{Q})}) = V(head_{\mathcal{Q}})$.

(only-if) Suppose there is an injective valuation V for $\mathcal{Q}$ and a valuation V' for $\mathcal{Q}$, such that $V' <_{\mathcal{Q}} V$. Then, $h \stackrel{\text{def}}{=} (V^{-1} \circ V')$ is a homomorphism from $\mathcal{Q}$ to itself, as $body_{h(\mathcal{Q})} = V^{-1}(V'(body_{\mathcal{Q}})) \subsetneq V^{-1}(V(body_{\mathcal{Q}})) = body_{\mathcal{Q}}$, $head_{h(\mathcal{Q})} = V^{-1}(V'(head_{\mathcal{Q}})) = V^{-1}(V(head_{\mathcal{Q}})) = head_{\mathcal{Q}}$. Therefore $h(\mathcal{Q})$ is equivalent to $\mathcal{Q}$, thanks to the homomorphism theorem (see, e.g., [1]). $\square$

Lemma 3.3 immediately yields the following complexity result.

PROPOSITION 3.4. *Deciding whether a valuation V for a CQ $\mathcal{Q}$ is minimal is* coNP*-complete.*

PROOF (SKETCH). Lemma 3.3 allows a reduction from minimality of CQs to minimality of valuations. Therefore, coNP-hardness follows from the coNP-hardness of minimality for CQs, which follows from [10]. The upper bound is immediate from the definition of minimality of valuations and from the fact that, for given $V_1, V_2, \mathcal{Q}$, it can be tested in polynomial time whether $V_1 <_{\mathcal{Q}} V_2$ holds. $\square$

Now, we are ready to settle the complexity of parallel-correctness for general conjunctive queries for a large class of distributions. We study two settings, $\mathcal{P}_{\text{fin}}$, where distribution policies are explicitly enumerated as part of the input, and $\mathcal{P}_{\text{nrel}}$, where the distribution policy is given by a black box procedure which answers questions of the form "$\kappa \in \boldsymbol{P}(\boldsymbol{f})$?" in NP. In the latter case, the distribution is not part of the (normal) input and therefore does not contribute to the input size. Instead, the input has an additional parameter n which bounds the length of addresses in the considered networks.

By $\mathbf{dom}_n$ we denote the set of all elements of $\mathbf{dom}$ that can be encoded by strings of length at most n. For a distribution policy $\boldsymbol{P}$ (coming with a network $\mathcal{N}$) and a number n, we denote by $\boldsymbol{P}_n$ the distribution policy that is obtained from $\boldsymbol{P}$ by (1) only distributing facts over $\mathbf{dom}_n$ and (2) only distributing facts to nodes whose addresses are of length at most n.

We study the following algorithmic problems for explicitly given database instances:

PCI($\mathcal{P}_{\text{fin}}$):
Input: CQ $\mathcal{Q}$, instance I, and $\boldsymbol{P} \in \mathcal{P}_{\text{fin}}$
Question: Is $\mathcal{Q}$ parallel-correct on I under $\boldsymbol{P}$?

PCI($\mathcal{P}_{\text{nrel}}$):
Input: CQ $\mathcal{Q}$, instance I, a natural number n in unary representation
Black box input: $\boldsymbol{P} \in \mathcal{P}_{\text{nrel}}$
Question: Is $\mathcal{Q}$ parallel-correct on I under $\boldsymbol{P}_n$?

We also study the parallel correctness problem without reference to a given database instance.

PC($\mathcal{P}_{\text{fin}}$):
Input: CQ $\mathcal{Q}$, $\boldsymbol{P} \in \mathcal{P}_{\text{fin}}$
Question: Is $\mathcal{Q}$ parallel-correct on I under $\boldsymbol{P}$, for all instances $I \subseteq facts(\boldsymbol{P})$?

Here, $facts(\boldsymbol{P})$ denotes the set of facts $\boldsymbol{f}$ with $\boldsymbol{P}(\boldsymbol{f}) \neq \emptyset$.

PC($\mathcal{P}_{\text{nrel}}$):
Input: CQ $\mathcal{Q}$, a natural number n in unary representation
Black box input: $\boldsymbol{P} \in \mathcal{P}_{\text{nrel}}$
Question: Is $\mathcal{Q}$ parallel-correct on I under $\boldsymbol{P}_n$, for all instances $I \subseteq facts(\boldsymbol{P}_n)$?

We quickly discuss how to use distribution policies from $\mathcal{P}_{\text{nrel}}$.

A distribution policy $\boldsymbol{P} \in \mathcal{P}_{\text{nrel}}$ is an NP-testable relation. This means that there exists a (deterministic) algorithm $\mathcal{A}_{\boldsymbol{P}}$ with time bound a polynomial in $\langle \kappa, \boldsymbol{f} \rangle$ that accepts input $(\langle \kappa, \boldsymbol{f} \rangle, x)$ for some string x if and only if $\kappa \in \boldsymbol{P}(\boldsymbol{f})$. We use algorithm $\mathcal{A}_{\boldsymbol{P}}$ as a subroutine in the following algorithms, as described below.

REMARK 3.5 (USE OF SUBROUTINE). *Let V be a valuation for a query $\mathcal{Q}$ with k body atoms and let κ be a node. We assume some additional input string $x = x_1 \circ \cdots \circ x_k$, where each substring x_i has a length polynomial in $V(body_{\mathcal{Q}})$ and the representation size of κ. An algorithm can "test" (w.r.t. x) whether there is a fact in $V(body_{\mathcal{Q}}) = \{\boldsymbol{f}_1, \ldots, \boldsymbol{f}_\ell\}$ that is* not *assigned to node κ under distribution policy $\boldsymbol{P}$, where $\ell \leq k$. To this end, the algorithm invokes $\mathcal{A}_{\boldsymbol{P}}$ as a subroutine with inputs $(\langle \kappa, \boldsymbol{f}_i \rangle, x_i)$ for each $i \in \{1, \ldots, \ell\}$. If any input is rejected, the algorithm accepts, otherwise it rejects. The running time is obviously bounded by the size of $V(body_{\mathcal{Q}})$ and the represention size of κ.*

THEOREM 3.6.

(a) $\mathrm{PC}(\mathcal{P}_{fin})$ and $\mathrm{PCI}(\mathcal{P}_{fin})$ are Π_2^P-complete.

(b) $\mathrm{PC}(\mathcal{P}_{nrel})$ and $\mathrm{PCI}(\mathcal{P}_{nrel})$ are in Π_2^P.

Of course, the upper bounds in Theorem 3.6 also hold if questions of the form "$\kappa \in \boldsymbol{P}(\boldsymbol{f})$?" are answered in polynomial time, or if $\boldsymbol{P}$ is just given as a polynomial time function.

Due to the implicit representation of distributions, we cannot formally claim Π_2^p-hardness for distribution policies from $\mathcal{P}_{\text{nrel}}$. However, in an informal sense, they are, of course, at least as difficult as for $\mathcal{P}_{\text{fin}}$.

PROOF (SKETCH). The upper bounds follow quite directly from Definition 2, or Lemma 3.1 and Proposition 3.4, respectively.

For the lower bound of $\mathrm{PCI}(\mathcal{P}_{\text{fin}})$ we give a polynomial reduction from the Π_2^P-complete problem Π_2-QBF, which can be adapted for $\mathrm{PC}(\mathcal{P}_{\text{fin}})$.

Let φ be an input for Π_2-QBF, i.e., a formula of the form $\forall \mathbf{x} \exists \mathbf{y}\, \psi(\mathbf{x}, \mathbf{y})$. We assume ψ to be a propositional formula in 3-CNF with variables $\mathbf{x} = (x_1, \ldots, x_m)$ and $\mathbf{y} = (y_1, \ldots, y_n)$. Let $C_1, \ldots, C_k$ denote their (disjunctive) clauses, where, for each j, $C_j = (\ell_{j,1} \vee \ell_{j,2} \vee \ell_{j,3})$.

We describe next how the corresponding input instance for $\mathrm{PCI}(\mathcal{P}_{\text{fin}})$, consisting of a query $\mathcal{Q}_\varphi$, a database instance I_φ, and a distribution policy $\boldsymbol{P}_\varphi$, is defined.

The query $\mathcal{Q}_\varphi$ is formulated over variables w_1, w_0, and $x_g, \overline{x}_g, y_h, \overline{y}_h$, for $g \in \{1, \ldots, m\}$ and $h \in \{1, \ldots, n\}$. Intuitively, these variables are intended to represent the Boolean values true and false and the (negated) values of the variables x_g, y_h in ψ, respectively. We overload the notation $\ell_{j,i}$ as follows: if $\ell_{j,i}$ is a negated literal $\neg x$ in C_j, then $\ell_{j,i}$ also denotes the variable $\overline{x}$.

Let $\mathbb{B}^+ \stackrel{\text{def}}{=} \mathbb{B} \setminus \{(0,0,0)\}$ be the set of non-zero Boolean triples and $\mathbb{W}^+ \stackrel{\text{def}}{=} \mathbb{W} \setminus \{(w_0, w_0, w_0)\}$ the set of triples over $\{w_0, w_1\}$ that contain at least one w_1.

We define $\mathcal{Q}_\varphi$ as the query with $head_{\mathcal{Q}_\varphi} = H(x_1, \ldots, x_m)$ and $body_{\mathcal{Q}_\varphi} = Cons \cup Struct(\psi)$, where

$$\begin{aligned} Cons &\stackrel{\text{def}}{=} \{\mathtt{True}(w_1), \mathtt{False}(w_0)\} \\ &\cup \{\mathtt{Neg}(w_1, w_0), \mathtt{Neg}(w_0, w_1)\} \\ &\cup \{\mathtt{C}_j(\mathbf{w}) \mid j \in \{1, \ldots, k\}, \mathbf{w} \in \mathbb{W}^+\} \end{aligned}$$

is a set of consistency atoms, representing valid combinations of values for $\mathtt{Neg}$-facts and satisfying combinations of values for $\mathtt{C}_j$-facts, and

$$\begin{aligned} Struct(\psi) &\stackrel{\text{def}}{=} \{\mathtt{Neg}(x, \overline{x}) \mid x \in \{x_1, \ldots, x_m, y_1, \ldots, y_n\}\}, \\ &\cup \{\mathtt{C}_j(\ell_{j,1}, \ell_{j,2}, \ell_{j,3}) \mid \\ &\qquad \text{for each clause } C_j = (\ell_{j,1} \vee \ell_{j,2} \vee \ell_{j,3})\} \end{aligned}$$

is a set of atoms representing the logical structure of ψ: it relates variable x_g to $\overline{x}_g$ and also variable y_h to $\overline{y}_h$ for each $g \in \{1, \ldots, m\}$ and $h \in \{1, \ldots, n\}$, respectively. Additionally, it relates all variables that represent literals occurring in the same clause to each other. Furthermore, we define

$$\begin{aligned} I_\varphi &\stackrel{\text{def}}{=} \{\mathtt{True}(1), \mathtt{False}(0), \mathtt{Neg}(1,0), \mathtt{Neg}(0,1)\} \\ &\cup \{\mathtt{C}_j(\mathbf{b}) \mid j \in \{1, \ldots, k\}, \mathbf{b} \in \mathbb{B}\}, \end{aligned}$$

which we partition into $I_\varphi^- \stackrel{\text{def}}{=} \{\mathtt{C}_j(0,0,0) \mid j \in \{1, \ldots, k\}\}$ and $I_\varphi^+ \stackrel{\text{def}}{=} I_\varphi \setminus I_\varphi^-$.

Moreover, we define $\boldsymbol{P}_\varphi$ to be the finite distribution policy for I_φ over a network $\mathcal{N} = \{\kappa^+, \kappa^-\}$ as

$$\boldsymbol{P}_\varphi(\boldsymbol{f}) = \begin{cases} \{\kappa^+\} & \text{if } \boldsymbol{f} \in I_\varphi^+, \\ \{\kappa^-\} & \text{if } \boldsymbol{f} \in I_\varphi^-. \end{cases}$$

It remains to show that this mapping is a polynomial-time reduction. Obviously, query $\mathcal{Q}_\varphi$, instance I_φ and distribution policy $\boldsymbol{P}_\varphi$ can be computed in polynomial time from φ. □

4. TRANSFERABILITY

Although parallel-correctness provides a direct one-round evaluation algorithm, it still requires a reshuffling of the data for every query. It therefore makes sense, in the context of multiple query evaluation, to consider scenarios in which such reshuffling can be avoided. To this end, we introduce the notion of parallel-correctness transfer which ensures that a subsequent query Q' can always be evaluated over a distribution for which a query Q is parallel-correct:

Definition 5. For two queries $\mathcal{Q}$ and $\mathcal{Q}'$ over the same input and output schema, *parallel-correctness transfers from $\mathcal{Q}$ to $\mathcal{Q}'$* when $\mathcal{Q}'$ is parallel-correct under every distribution policy for which $\mathcal{Q}$ is parallel-correct.

As for parallel-correctness we first give a semantical characterization before we study the complexity of parallel-correctness transfer.

LEMMA 4.1. *Parallel-correctness transfers from a CQ $\mathcal{Q}$ to a CQ $\mathcal{Q}'$ if and only if the following holds:*

(C2) for every minimal valuation V' for $\mathcal{Q}'$, there is a minimal valuation V for $\mathcal{Q}$ with $V'(body_{\mathcal{Q}'}) \subseteq V(body_{\mathcal{Q}})$.

The two implications of Lemma 4.1 are shown in Propositions 4.2 and 4.3 below.

PROPOSITION 4.2. *Let $\mathcal{Q}$ and $\mathcal{Q}'$ be CQs. If condition (C2) holds, then, parallel-correctness transfers from $\mathcal{Q}$ to $\mathcal{Q}'$.*

PROOF. Let $\boldsymbol{P}$ be a distribution policy under which $\mathcal{Q}$ is parallel-correct and let I be an instance. Then we show that $\mathcal{Q}'$ is parallel-correct as well on I under $\boldsymbol{P}$. By monotonicity of CQs, $\bigcup_{x \in \mathcal{N}} \mathcal{Q}'(dist_{\boldsymbol{P},I}(x)) \subseteq \mathcal{Q}'(I)$. Thus it suffices to show that for every fact $\boldsymbol{f} \in \mathcal{Q}'(I)$, there is some valuation for $\mathcal{Q}'$ that allows to derive $\boldsymbol{f}$ on one of the chunks of I under $\boldsymbol{P}$. For $\boldsymbol{f} \in \mathcal{Q}'(I)$, there is a minimal valuation V' for $\mathcal{Q}'$ which satisfies on I for $\mathcal{Q}'$ and derives $\boldsymbol{f}$. That is, $V'(body_{\mathcal{Q}'}) \subseteq I$ and $V'(head_{\mathcal{Q}'}) = \boldsymbol{f}$. Next, we show that the facts required by V' for $\mathcal{Q}'$ meet at some node under $\boldsymbol{P}$, which implies that the chunks of I under $\boldsymbol{P}$ indeed allow deriving $\boldsymbol{f}$.

For this, we rely on the assumption that there is a minimal valuation V for $\mathcal{Q}$, where $V'(body_{\mathcal{Q}'}) \subseteq V(body_{\mathcal{Q}})$. Let $J = V(body_{\mathcal{Q}})$. Then, by parallel-correctness of $\mathcal{Q}$ under $\boldsymbol{P}$, there is a valuation W and node $\kappa \in \mathcal{N}$, such that $W(body_{\mathcal{Q}}) \subseteq dist_{\boldsymbol{P},J}(\kappa)$ and $W(head_{\mathcal{Q}}) = V(head_{\mathcal{Q}})$. Because V is minimal and $dist_{\boldsymbol{P},J}(\kappa) \subseteq V(body_{\mathcal{Q}})$, it must be that $V(body_{\mathcal{Q}}) = W(body_{\mathcal{Q}})$. So, $\boldsymbol{P}$ maps all the facts in J onto node κ, implying that all the facts in $V'(body_{\mathcal{Q}'})$ are mapped onto node κ under $\boldsymbol{P}$ (because $V'(body_{\mathcal{Q}'}) \subseteq V(body_{\mathcal{Q}}) = J$).

Hence, $\mathcal{Q}'$ is indeed parallel-correct under the distribution policies for which $\mathcal{Q}$ is parallel-correct. $\square$

PROPOSITION 4.3. *Let $\mathcal{Q}$ and $\mathcal{Q}'$ be CQs. If parallel-correctness transfers from $\mathcal{Q}$ to $\mathcal{Q}'$, then, condition (C2) holds.*

PROOF. The proof is by contraposition. So, we assume that there is a minimal valuation V' for $\mathcal{Q}'$ for which there is no valuation V for $\mathcal{Q}$, where $V'(body_{\mathcal{Q}'}) \subseteq V(body_{\mathcal{Q}})$. Let $m = |V'(body_{\mathcal{Q}'})|$.

We distinguish two cases, depending on whether V' requires only one fact or at least two facts. For both cases we construct a network $\mathcal{N}$ and distribution policy $\boldsymbol{P}$ over $\mathcal{N}$ for which $\mathcal{Q}$ is parallel-correct but $\mathcal{Q}'$ is not, implying that parallel-correctness does *not* transfer from $\mathcal{Q}$ to $\mathcal{Q}'$.

(Case $m = 1$) Let $V'(body_{\mathcal{Q}'}) = \{\boldsymbol{f}\}$. Let $\mathcal{N}$ be a single-node network, i.e., $\mathcal{N} \stackrel{\text{def}}{=} \{\kappa\}$. For $\boldsymbol{P}$ we consider the distribution policy thats skips $\boldsymbol{f}$, that is, maps $\boldsymbol{P}(\boldsymbol{f})$ to the empty set, and maps every other fact in $facts(\mathcal{D})$ onto node κ. By assumption on V', none of the minimal valuations for $\mathcal{Q}$ requires $\boldsymbol{f}$. So it immediately follows by Lemma 3.1 that $\mathcal{Q}$ is parallel-correct under $\boldsymbol{P}$. However, because V' is minimal for $\mathcal{Q}'$, $\mathcal{Q}'$ needs $\boldsymbol{f}$ to derive $V(head_{\mathcal{Q}'})$ when only $\boldsymbol{f}$ is given as input instance. Thus $\mathcal{Q}'$ is not parallel-correct under $\boldsymbol{P}$ which leads to the desired contradiction.

(Case $m \geq 2$) Let $I \stackrel{\text{def}}{=} V'(body_{\mathcal{Q}'}) = \{\boldsymbol{f}_1, \ldots, \boldsymbol{f}_m\}$, $\mathcal{N} \stackrel{\text{def}}{=} \{\kappa_1, \ldots, \kappa_m\}$, and let $\boldsymbol{P}$ be the mapping defined as follows:

- $\boldsymbol{P}(\boldsymbol{g}) = \mathcal{N}$, for every $\boldsymbol{g} \in facts(\mathcal{D}) \setminus I$; and
- $\boldsymbol{P}(\boldsymbol{f}_i) = \mathcal{N} \setminus \{\kappa_i\}$, for every i.

Intuitively, on every instance J, either the facts in J meet on some node under $\boldsymbol{P}$, or $I \subseteq J$. By assumption, none of the minimal valuations for $\mathcal{Q}$ requires all the facts in I, implying that $\mathcal{Q}$ is parallel-correct under $\boldsymbol{P}$. Nevertheless, on instance I under $\boldsymbol{P}$, none of the nodes receives all the facts in I, and there is no valuation that can derive $V'(head_{\mathcal{Q}'})$ for a strict subset of the facts in I (by minimality of V'). So, $\mathcal{Q}'$ is not parallel-correct under $\boldsymbol{P}$ which leads to the desired contradiction. $\square$

The characterisation given by Lemma 4.1 allows us to pinpoint the complexity of parallel-correctness transfers. For a formal statement we define the following algorithmic problem:

PC-TRANS:
Input: CQs $\mathcal{Q}$ and $\mathcal{Q}'$
Question: Does parallel-correctness transfer from $\mathcal{Q}$ to $\mathcal{Q}'$?

In principle, Lemma 4.1, on which the following proofs are based, talks about an infinite number of valuations over the infinite domain **dom**. However, since our queries are generic, the only observable property of the constants used by some valuation is equality/inequality. It therefore suffices to check valuations over an arbitrary finite domain with at least as much constants as valuations for both queries can use. This is stated more explicitly in the following claim.

CLAIM 1. *Let $\mathcal{Q}$ and $\mathcal{Q}'$ be CQs with variables $x_1, \ldots, x_m$ and $y_1, \ldots, y_n$, respectively. Moreover, for $k = m + n$ let $dom_k = \{1, \ldots, k\}$ be a subset of the (countably) infinite set* **dom**.

The following two conditions are equivalent.

1. *For every minimal valuation V' for $\mathcal{Q}'$ over* **dom** *there is a minimal valuation V for $\mathcal{Q}$ over* **dom** *such that $V'(body_{\mathcal{Q}'}) \subseteq V(body_{\mathcal{Q}})$.*
2. *For every minimal valuation V' for $\mathcal{Q}'$ over dom_k there is a minimal valuation V for $\mathcal{Q}$ over dom_k such that $V'(body_{\mathcal{Q}'}) \subseteq V(body_{\mathcal{Q}})$.*

THEOREM 4.4. PC-TRANS *is Π_3^P-complete.*

PROOF (SKETCH). For the upper bound, we note that, by Lemma 4.1, deciding parallel-correctness transfer is equivalent to verifying that for each minimal valuation V' for $\mathcal{Q}'$ there is a minimal valuation V for $\mathcal{Q}$ such that $V'(body_{\mathcal{Q}'}) \subseteq V(body_{\mathcal{Q}})$. This, in turn, is equivalent to checking for each valuation V' for $\mathcal{Q}'$ that it is not minimal, which can be witnessed by another valuation W' that derives the same fact and requires strictly less facts, *or* that there is a minimal valuation V for $\mathcal{Q}$ such that $V'(body_{\mathcal{Q}'_\varphi}) \subseteq V(body_{\mathcal{Q}_\varphi})$. Non-minimality of valuation V can be witnessed by a valuation W. Thanks to Claim 1, all valuations can be restricted to $dom_k = \{1, \ldots, k\}$, where $k = m+n$ and $\mathcal{Q}, \mathcal{Q}'$ are queries over variables $x_1, \ldots, x_m$ and $y_1, \ldots, y_n$, respectively.

To prove membership in class Π_3^P, it suffices to show that there is an algorithm with a time bound polynomial in $|\mathcal{Q}| + |\mathcal{Q}'|$ such that for every pair $(\mathcal{Q}, \mathcal{Q}')$ of queries it holds $(\mathcal{Q}, \mathcal{Q}') \in$ PC-TRANS if and only if for every $\mathcal{Q}'$-valuation V' there is a $\mathcal{Q}$-valuation V and a $\mathcal{Q}'$-valuation W' such that for every $\mathcal{Q}$-valuation W, the algorithm accepts $\big(\langle \mathcal{Q}, \mathcal{Q}' \rangle, V', \langle V, W' \rangle, W\big)$.

For input $\big(\langle \mathcal{Q}, \mathcal{Q}' \rangle, V', \langle V, W' \rangle, W\big)$ the algorithm proceeds as follows. First, it is checked whether W' contradicts the assumed minimality of V', that is, whether $W'(head_{\mathcal{Q}'}) = V'(head_{\mathcal{Q}'})$ as well as $W'(body_{\mathcal{Q}'}) \subsetneq V'(body_{\mathcal{Q}'})$. If this test succeeds, the algorithm accepts because there is no requirement on a non-minimal $\mathcal{Q}'$-valuation. Second, it is checked in an analogous fashion whether W contradicts the assumed minimality of V. If this test succeeds, the algorithm rejects.

Lastly, the algorithm continues with testing $V'(body_{\mathcal{Q}'}) \subseteq V(body_{\mathcal{Q}})$. It accepts in case of satisfaction, and rejects otherwise. All containment tests can be done in polynomial time.

The lower bound is by a reduction from the Π_3^P-complete Π_3-QBF-problem. The reduction is based on the characterization of parallel-correctness transfer by condition (C2) as stated in Lemma 4.1.

Reduction function. Let $\varphi = \forall \mathbf{x} \exists \mathbf{y} \forall \mathbf{z}\, \psi(\mathbf{x}, \mathbf{y}, \mathbf{z})$ be a formula with a quantifier-free propositional formula ψ in 3-DNF over variables $\mathbf{x} = (x_1, \ldots, x_m)$, $\mathbf{y} = (y_1, \ldots, y_n)$, and $\mathbf{z} = (z_1, \ldots, z_p)$.

Let k be the number of clauses of ψ and, for each $j \in \{1, \ldots, k\}$, let $C_j = (\ell_{j,1} \wedge \ell_{j,2} \wedge \ell_{j,3})$ denote the j-th (conjunctive) clause of ψ.

The reduction function maps φ to a pair $(\mathcal{Q}_\varphi, \mathcal{Q}'_\varphi)$ of CQs that will be described next. It will be obvious that this mapping can be computed in polynomial time. Query $\mathcal{Q}_\varphi$ uses the variables w_1, w_0, which are intended to represent truth and falseness, respectively, the variables of ψ and variables $\overline{u}$, for each variable u of ψ, representing the literal $\neg u$.[5] Besides these variables, query $\mathcal{Q}'_\varphi$ additionally uses the following variables

- s_j, for every $j \in \{1, \ldots, k\}$, intended to represent the truth value of C_j, and
- r_j, for every $j \in \{1, \ldots, k\}$, intended to represent the truth value of $C_1 \vee \cdots \vee C_j$.

We first describe the general construction, give an example explaining its intuition afterwards and finally prove correctness of the reduction.

The queries $\mathcal{Q}_\varphi$ and $\mathcal{Q}'_\varphi$ are defined as follows:

$$\begin{aligned}
\mathit{head}_{\mathcal{Q}_\varphi} &\stackrel{\text{def}}{=} H(x_1, \ldots, x_m, w_1, w_0)\\
\mathit{body}_{\mathcal{Q}_\varphi} &\stackrel{\text{def}}{=} \big\{\texttt{YVal}_h(w_1), \texttt{YVal}_h(w_0) \mid h \in \{1, \ldots, n\}\big\}\\
&\cup \quad \{\texttt{Res}(w_1)\} \cup \mathit{Fix};
\end{aligned}$$

$$\begin{aligned}
\mathit{head}_{\mathcal{Q}'_\varphi} &\stackrel{\text{def}}{=} H(x_1, \ldots, x_m, y_1, \ldots, y_n, w_1, w_0)\\
\mathit{body}_{\mathcal{Q}'_\varphi} &\stackrel{\text{def}}{=} \big\{\texttt{YVal}_h(y_h), \texttt{YVal}_h(\overline{y}_h) \mid h \in \{1, \ldots, n\}\big\}\\
&\cup \quad \{\texttt{Res}(w_0), \texttt{Res}(r_k)\} \cup \mathit{Fix} \cup \mathit{Gates} \cup \mathit{Circuit},
\end{aligned}$$

where

$$\mathit{Fix} \stackrel{\text{def}}{=} \big\{\texttt{XVal}_1(x_1), \ldots, \texttt{XVal}_m(x_m), \texttt{True}(w_1), \texttt{False}(w_0)\big\}$$

is intended to "fix" truth values for $x_1, \ldots, x_m, w_1, w_0$, while the set

$$\begin{aligned}
\mathit{Gates} \stackrel{\text{def}}{=} &\ \{\texttt{Neg}(w_0, w_1), \texttt{Neg}(w_1, w_0)\}\\
\cup &\ \{\texttt{And}(w_1, w_1, w_1, w_1), \texttt{And}(w_0, w_1, w_1, w_0),\\
&\ \ \texttt{And}(w_1, w_0, w_1, w_0), \texttt{And}(w_0, w_0, w_1, w_0),\\
&\ \ \texttt{And}(w_1, w_1, w_0, w_0), \texttt{And}(w_0, w_1, w_0, w_0),\\
&\ \ \texttt{And}(w_1, w_0, w_0, w_0), \texttt{And}(w_0, w_0, w_0, w_0)\}\\
\cup &\ \{\texttt{Or}(w_1, w_1, w_1), \texttt{Or}(w_0, w_1, w_1),\\
&\ \ \texttt{Or}(w_1, w_0, w_1), \texttt{Or}(w_0, w_0, w_0)\}
\end{aligned}$$

contains all atoms that are consistent with respect to the intended meaning of negation, And- and Or-gates[6] on w_1, w_0, and

$$\begin{aligned}
\mathit{Circuit} \stackrel{\text{def}}{=} &\ \{\texttt{Neg}(u, \overline{u}) \mid \text{for each variable } u \text{ in } \psi\}\\
\cup &\ \{\texttt{And}(\ell_{j,1}, \ell_{j,2}, \ell_{j,3}, s_j) \mid\\
&\qquad \text{for each clause } C_j = (\ell_{j,1} \wedge \ell_{j,2} \wedge \ell_{j,3})\}\\
\cup &\ \{\texttt{Or}(s_1, s_1, r_1)\}\\
\cup &\ \{\texttt{Or}(r_1, s_2, r_2), \ldots, \texttt{Or}(r_{k-1}, s_k, r_k)\}
\end{aligned}$$

is intended to represent a Boolean circuit (with output bit r_k) that evaluates ψ.

[5] If ℓ is a negated literal $\neg u$, we write ℓ also for $\overline{u}$.

[6] The last position in a gate-atom represents the output bit of the gate, the others the input bits.

Example 4.5. *We obtain the queries displayed in Figure 1 for $\varphi = \forall x_1 \exists y_1 \exists y_2 \forall z_1 \Big((x_1 \wedge y_1 \wedge z_1) \vee (\neg x_1 \wedge y_2 \wedge z_1)\Big)$.*

Note that $\varphi \notin \Pi_3$-QBF because no truth assignment with $z_1 \mapsto 0$ is satisfying for ψ. In particular, for the truth assignment $\beta_\mathbf{x} : x_1 \mapsto 1$ there is no truth assignment $\beta_\mathbf{y}$ such that for every $\beta_\mathbf{z}$ it holds $(\beta_\mathbf{x} \cup \beta_\mathbf{y} \cup \beta_\mathbf{z}) \models \psi$. We illustrate why $(\mathcal{Q}_\varphi, \mathcal{Q}'_\varphi) \notin$ PC-TRANS.

Let valuation V for $\mathcal{Q}_\varphi$ be defined by $V(x_1) \stackrel{\text{def}}{=} \beta_\mathbf{x}(x_1) = 1$, $V(w_1) \stackrel{\text{def}}{=} 1$ and $V(w_0) \stackrel{\text{def}}{=} 0$. This valuation is minimal for $\mathcal{Q}_\varphi$ (because $\mathcal{Q}_\varphi$ is full) and requires the set $V(\mathit{body}_{\mathcal{Q}_\varphi}) = \big\{\texttt{YVal}_1(1), \texttt{YVal}_1(0), \texttt{Res}(1), \texttt{XVal}_1(1), \texttt{True}(1), \texttt{False}(0)\big\}$ of facts.

We now argue why there is no minimal valuation V' for $\mathcal{Q}'_\varphi$ such that $V'(\mathit{body}_{\mathcal{Q}'_\varphi}) \subseteq V(\mathit{body}_{\mathcal{Q}_\varphi})$. If a valuation V' fulfills $V'(\mathit{body}_{\mathcal{Q}'_\varphi}) \subseteq V(\mathit{body}_{\mathcal{Q}_\varphi})$ it must map $w_0 \mapsto 0$, $w_1 \mapsto 1$, $x_1 \mapsto 1$, $r_2 \mapsto 1$. Furthermore, it must map each of $(y_1, \overline{y}_1)$ and $(y_2, \overline{y}_2)$ to some pair in $\{(0,1), (1,0)\}$. Thus, V induces a truth assignment $\beta_\mathbf{y}$ via $\beta_\mathbf{y}(y_1) \stackrel{\text{def}}{=} V(y_1)$ and $\beta_\mathbf{y}(y_2) \stackrel{\text{def}}{=} V(y_2)$. Let V^* be the valuation that coincides with V on all variables $w_0, w_i, x_1, \overline{x}_1, y_1, \overline{y}_1, y_2, \overline{y}_2$ and maps $z_1 \mapsto 0$ and maps all other variables to the "correct" values with respect to the semantics of the logical gates in $\mathcal{Q}_\varphi$. In particular, since $(\beta_\mathbf{x} \cup \beta_\mathbf{y} \cup \beta_\mathbf{z}) \not\models \psi$ (where $\beta_\mathbf{z}(z_1) \stackrel{\text{def}}{=} 0$), we get $V^*(r_2) = 0$. It is now easy to check that $V^* <_\mathcal{Q} V$, and therefore that V is not minimal.

To complete the proof, we need to show that the mapping $\varphi \mapsto (\mathcal{Q}_\varphi, \mathcal{Q}'_\varphi)$ is indeed a reduction, i.e. that φ is in Π_3-QBF if and only if parallel-correctness transfers from $\mathcal{Q}_\varphi$ to $\mathcal{Q}'_\varphi$.

We start by some observations. We call a valuation for $\mathcal{Q}_\varphi$ or $\mathcal{Q}'_\varphi$ *0-1-valued*, if its range is $\{0,1\}$ and it maps (w_0, w_1) to $(0,1)$ and every pair $(u, \overline{u})$ of variables from ψ to $(0,1)$ or $(1,0)$. A 0-1-valued valuation is called *consistent*, if the values $V(s_j)$ and $V(r_j)$, for $j \in \{1, \ldots, k\}$ are consistent with the values $V(u)$ for variables of ψ, in the obvious sense. That is, $V(s_j) = 1$ if and only if clause C_j evaluates to true for the truth assignment β_V obtained from V and $V(r_j) = 1$ if and only if $C_1 \vee \cdots \vee C_j$ evaluates to true.

It is easy to see that a 0-1-valued valuation V is consistent, if and only if $V(\mathit{Circuit}) \subseteq V(\mathit{Gates})$, because inconsistency requires facts in $V(\mathit{Circuit})$ that are not in $V(\mathit{Gates})$ and likewise the existence of such facts implies inconsistency.

Claim 2. *For every 0-1-valued valuation V of $\mathcal{Q}_\varphi$ the following conditions are equivalent.*

(i) V is minimal;

(ii) V is consistent.

(only-if). Let $\varphi = \forall \mathbf{x} \exists \mathbf{y} \forall \mathbf{z}\, \psi(\mathbf{x}, \mathbf{y}, \mathbf{z})$ be a formula with a quantifier-free propositional formula ψ in 3-DNF such that $\varphi \notin \Pi_3$-QBF. We show that there is a minimal valuation V for $\mathcal{Q}_\varphi$ such that each valuation V' for $\mathcal{Q}'_\varphi$ which satisfies $V'(\mathit{body}_{\mathcal{Q}'_\varphi}) \subseteq V(\mathit{body}_{\mathcal{Q}_\varphi})$ is *not* minimal. From that we can conclude by Lemma 4.1 that parallel-correctness does not transfer from $\mathcal{Q}_\varphi$ to $\mathcal{Q}'_\varphi$.

Let $\beta_\mathbf{x}$ be a truth assignment for $x_1, \ldots, x_m$ in ψ such that for all truth assignments $\beta_\mathbf{y}$ for $y_1, \ldots, y_n$ in ψ there is a truth assignment $\beta_\mathbf{z}$ for $z_1, \ldots, z_p$ such that $(\beta_\mathbf{x} \cup \beta_\mathbf{y} \cup \beta_\mathbf{z}) \not\models \psi$.

$$\mathcal{Q}_\varphi: \quad H(x_1, w_1, w_0) \quad \leftarrow \quad \texttt{YVal}_1(w_1), \texttt{YVal}_1(w_0), \texttt{YVal}_2(w_1), \texttt{YVal}_2(w_0), \texttt{Res}(w_1), \texttt{XVal}_1(x_1), \texttt{True}(w_1), \texttt{False}(w_0).$$

$$\begin{aligned}\mathcal{Q}'_\varphi: \quad H(x_1, x_2, y_1, w_1, w_0) \quad \leftarrow \quad & \texttt{YVal}_1(y_1), \texttt{YVal}_1(\overline{y}_1), \texttt{YVal}_2(y_2), \texttt{YVal}_2(\overline{y}_2), \texttt{Res}(w_0), \texttt{Res}(r_2),\\ & \texttt{XVal}_1(x_1), \texttt{True}(w_1), \texttt{False}(w_0),\\ & \ldots \textit{all atoms from Gates} \ldots,\\ & \texttt{Neg}(x_1, \overline{x}_1), \texttt{Neg}(y_1, \overline{y}_1), \texttt{Neg}(y_2, \overline{y}_2), \texttt{Neg}(z_1, \overline{z}_1),\\ & \texttt{And}(x_1, y_1, z_1, s_1), \texttt{And}(\overline{x}_1, y_2, z_1, s_2), \texttt{Or}(s_1, s_1, r_1), \texttt{Or}(r_1, s_2, r_2).\end{aligned}$$

Figure 1: Output of the reduction function on input $\varphi = \forall x_1 \exists y_1 \exists y_2 \forall z_1 \; \big((x_1 \wedge y_1 \wedge z_1) \vee (\neg x_1 \wedge y_2 \wedge z_1)\big)$.

Let V be the valuation defined by $V(x_1, \ldots, x_m, w_1, w_0) \stackrel{\text{def}}{=} (\beta_\mathbf{x}(x_1), \ldots, \beta_\mathbf{x}(x_m), 1, 0)$, which is minimal for $\mathcal{Q}_\varphi$ because $\mathcal{Q}_\varphi$ is full.

Let V' be any valuation for $\mathcal{Q}'_\varphi$ such that $V'(\textit{body}_{\mathcal{Q}'_\varphi}) \subseteq V(\textit{body}_{\mathcal{Q}_\varphi})$. In particular, $\texttt{Res}(1) \in V'(\textit{body}_{\mathcal{Q}'_\varphi})$. Then, valuations V and V' agree on variables $x_1, \ldots, x_m, w_1, w_0$ because each atom in *Fix* is the only atom of $\mathcal{Q}_\varphi$ with its particular relation symbol. Similarly, the $\texttt{YVal}_i$-atoms in $\mathcal{Q}_\varphi$ and $\mathcal{Q}'_\varphi$ ensure that V' maps each pair $(y_i, \overline{y}_i)$ to $(0,1)$ or $(1,0)$. Let $\beta_\mathbf{y}$ be the truth assignment defined by $\beta_\mathbf{y}(y_i) \stackrel{\text{def}}{=} V(y_i)$, for every $i \in \{1, \ldots, n\}$. Since $\varphi \notin \Pi_3$-QBF, there is a truth assignment $\beta_\mathbf{z}$ such that $(\beta_\mathbf{x} \cup \beta_\mathbf{y} \cup \beta_\mathbf{z}) \not\models \psi$. Let V^* be the uniquely defined consistent 0-1-valued valuation induced by $(\beta_\mathbf{x} \cup \beta_\mathbf{y} \cup \beta_\mathbf{z})$. Since V^* is consistent, $V^*(\textit{Circuit}) \subseteq V^*(\textit{Gates})$ and therefore $V^*(\textit{body}_{\mathcal{Q}_\varphi}) \subseteq V(\textit{body}_{\mathcal{Q}_\varphi})$. Furthermore, since $(\beta_\mathbf{x} \cup \beta_\mathbf{y} \cup \beta_\mathbf{z}) \not\models \psi$, we get $V^*(r_k) = 0$ and therefore $\texttt{Res}(1) \notin V^*(\textit{body}_{\mathcal{Q}_\varphi})$ and, consequently, $V^*(\textit{body}_{\mathcal{Q}_\varphi}) \subsetneq V(\textit{body}_{\mathcal{Q}_\varphi})$, showing that V is not minimal.

(if). Let $\varphi = \forall \mathbf{x} \exists \mathbf{y} \forall \mathbf{z}\, \psi(\mathbf{x}, \mathbf{y}, \mathbf{z})$ be a formula in Π_3-QBF and let V' be an arbitrary valuation for $\mathcal{Q}'_\varphi$. We will show that there exists a minimal valuation V for $\mathcal{Q}_\varphi$ such that $V'(\textit{body}_{\mathcal{Q}'_\varphi}) \subseteq V(\textit{body}_{\mathcal{Q}_\varphi})$, and thus that parallel-correctness transfers from $\mathcal{Q}_\varphi$ to $\mathcal{Q}'_\varphi$, again by Lemma 4.1.

We assume in the following that all quantified variables appear (possibly negated) in ψ. Let $c_0 \stackrel{\text{def}}{=} V'(w_0)$ and $c_1 \stackrel{\text{def}}{=} V'(w_1)$. Since, neither $\mathcal{Q}'_\varphi$ nor $\mathcal{Q}_\varphi$ uses any constant symbols, minimality of V' is not affected, if V' is composed with any bijection of the domain. The same holds for every valuation V and the statement $V'(\textit{body}_{\mathcal{Q}'_\varphi}) \subseteq V(\textit{body}_{\mathcal{Q}_\varphi})$, as long as V' and V are composed with the *same* bijection. Therefore, we can assume without loss of generality that $V'(w_0) = 0$ and $V'(w_1) \in \{0,1\}$.

We distinguish between three cases depending on whether $\mathbf{dom}(V') \subseteq \{0,1\}$ and $V'(w_1) = 1$.

Case 1 ($\mathbf{dom}(V') \subseteq \{0,1\}$ and $V'(w_1) = 1$): Let $\beta_\mathbf{x}$ be the partial truth assignment for the variables $x_1, \ldots, x_m$ in ψ defined by $\beta_\mathbf{x}(x_i) = V'(x_i)$, for every $i \in \{1, \ldots, m\}$. Since, $\varphi \in \Pi_3$-QBF, there exists a partial truth assignment $\beta_\mathbf{y}$ for the variables $y_1, \ldots, y_n$ in ψ such that for each partial truth assignment $\beta_\mathbf{z}$ for the variables $z_1, \ldots, z_p$ we have $(\beta_\mathbf{x} \cup \beta_\mathbf{y} \cup \beta_\mathbf{z}) \models \psi$. For concreteness let $\beta_\mathbf{z}(z_i) \stackrel{\text{def}}{=} 0$, for $i \in \{1, \ldots, p\}$ and $\beta \stackrel{\text{def}}{=} \beta_\mathbf{x} \cup \beta_\mathbf{y} \cup \beta_\mathbf{z}$.

Let V be the uniquely defined 0-1-valued consistent valuation induced by β. Since V is consistent it is also minimal by Claim 2, and as $\beta \models \psi$, $V(r_k) = 1$. Thanks to the latter, $V'(\textit{body}_{\mathcal{Q}'_\varphi}) \subseteq V(\textit{body}_{\mathcal{Q}_\varphi})$ follows easily and Case 1 is complete.

Case 2 ($\mathbf{dom}(V') \subseteq \{0,1\}$ and $V'(w_1) = 0$): Let V be defined by

$$V(u) \stackrel{\text{def}}{=} \begin{cases} V'(u) & \text{if } u \in \{w_0, w_1, x_1, \ldots, x_n\}, \\ 0 & \text{otherwise.} \end{cases}$$

It is easy to see that $V'(\textit{body}_{\mathcal{Q}'_\varphi}) \subseteq V(\textit{body}_{\mathcal{Q}_\varphi})$. Furthermore, V' is minimal as every fact from $V(\textit{body}_{\mathcal{Q}_\varphi})$ either stems from an atom with (only) head variables or is in the unavoidable set $V(\textit{Gates})$.

Case 3 (For some g, $V'(x_g) \notin \{c_0, c_1\}$): We recall that by our assumptions, $c_0 = 0 = V'(w_0)$ and $c_1 = V'(w_1) \in \{0,1\}$. The following argument works for both subcases, $c_1 = 1$ and $c_1 = 0$. We call a variable x_g *foul* if $V'(x_g) \notin \{c_0, c_1\}$. Likewise, we call a clause *foul* if it contains (positively or negatively) some foul variable. Let G be the set of all indices g for which x_g is foul and J be the set of all indices j of foul clauses. Furthermore, let[7] $a = V'(x_g)$ for the minimal index $g \in G$.

We define valuation V by

$$V(u) \stackrel{\text{def}}{=} \begin{cases} V'(u) & \text{if } u \in \{w_0, w_1, x_1, \ldots, x_m\}, \\ c_1 & \text{if } u \in \{y_1, \ldots, y_n, z_1, \ldots, z_p\}, \\ c_0 & \text{if } u \in \{\overline{y}_1, \ldots, \overline{y}_n, \overline{z}_1, \ldots, \overline{z}_p\}, \\ a & \text{if } u = \overline{x}_g \text{ and } x_g \text{ is foul}, \\ c_0 & \text{if } u = \overline{x}_g \text{ and } V'(x_g) = c_1, \\ c_1 & \text{if } u = \overline{x}_g \text{ and } V'(x_g) = c_0. \end{cases}$$

For variables s_j, $V(s_j) \stackrel{\text{def}}{=} c_1$, if C_j is foul or for all its literals ℓ, it holds $V(\ell) = c_1$, otherwise $V(s_j) \stackrel{\text{def}}{=} c_0$. For variables r_j, $V(r_j) \stackrel{\text{def}}{=} c_1$, if $V(s_i) = c_1$, for some $i \leq j$ and $V(r_j) \stackrel{\text{def}}{=} c_0$, otherwise.

It is clear that $V'(\textit{body}_{\mathcal{Q}'_\varphi}) \subseteq V(\textit{body}_{\mathcal{Q}_\varphi})$ holds, but we can not expect that V is minimal. There might be some $\texttt{And}$-facts in $V(\textit{Circuit})$ resulting from clauses that can be avoided by changing the valuation for some variables z_i. However, we can show that every minimal valuation V^* contained in V fulfills $V'(\textit{body}_{\mathcal{Q}'_\varphi}) \subseteq V^*(\textit{body}_{\mathcal{Q}_\varphi})$ and thereby yields (C3).

To this end, let V^* be a minimal valuation such that $V^* \leq_{\mathcal{Q}} V$. We show first that V^* has to produce most facts from $V(\textit{body}_{\mathcal{Q}_\varphi})$. This is immediate for all facts from $V(\{\texttt{YVal}_h(y_h), \texttt{YVal}_h(\overline{y}_h) \mid h \in \{1, \ldots, n\}\})$, and also for those from $V(\textit{Fix})$, and $V(\textit{Gates})$.

[7] In fact, any value not in $\{c_0, c_1\}$ would do.

Any facts of the form $V(\texttt{Neg}(u,\overline{u}))$ that do not occur in $V(\mathit{Gates})$ are of the form $\texttt{Neg}(V'(x_g),a)$, for some foul variable x_g. As x_g occurs in the head, and there is at most one such fact per foul variable, these facts can not be avoided in $V^*(\mathit{body}_{\mathcal{Q}_\varphi})$. As all facts of the form $V^*(\texttt{Neg}(u,\overline{u}))$ have to be in $V^*(\mathit{body}_{\mathcal{Q}_\varphi})$ and all variables x_i, y_i occur in $\mathit{head}_{\mathcal{Q}_\varphi}$, we can conclude that V^* has to agree with V for all variables of the form $x_i, \overline{x}_i, y_i, \overline{y}_i$ and on w_0 and w_1.

Therefore, it is clear for all facts from $V'(\mathit{body}_{\mathcal{Q}'_\varphi})$ except $\texttt{Res}(c_1)$ that they are captured by $V^*(\mathit{body}_{\mathcal{Q}_\varphi})$. It therefore only remains to show $\texttt{Res}(c_1) \in V^*(\mathit{body}_{\mathcal{Q}_\varphi})$.

Let x_g be the foul variable that was used to define $a \stackrel{\text{def}}{=} V'(x_g)$ and let C_j be some clause in which it occurs. Thus, by definition of V there is an $\texttt{And}$-fact in $V(\mathit{Circuit})$ with value a in one of its first three positions and with c_1 in its fourth position. Furthermore, all $\texttt{And}$-facts in $V(\mathit{Circuit})$ with a-values have c_1 in their fourth position. Therefore, $V^*(\mathit{Circuit})$ needs to contain at least one $\texttt{And}$-fact with a in one of its first three positions and with c_1 in its fourth position. That is, $V^*(s_i) = c_1$, for at least one i. As $V^*(\mathit{Circuit})$ can only contain $\texttt{Or}$-facts from $V(\mathit{Gates})$, it follows that $V^*(r_h) = c_1$, for all $h \geq i$ and, in particular, for $h = k$. Therefore, $\texttt{Res}(c_1) \in V^*(\mathit{body}_{\mathcal{Q}_\varphi})$ and $V'(\mathit{body}_{\mathcal{Q}'_\varphi}) \subseteq V^*(\mathit{body}_{\mathcal{Q}_\varphi})$. $\square$

It is an easy observation that, if we require each valuation of $\mathcal{Q}$ to be minimal, then condition (C2) yields a better, Π_2^P, complexity bound. Surprisingly, in this case, we even get a complexity drop to NP, as will be shown in Theorem 4.8 below. We next introduce the notions needed for this result.

Definition 6. A conjunctive query $\mathcal{Q}$ is *strongly minimal* if all its valuations are minimal.

We give some examples illustrating this definition. In Lemma 4.9, we present a sufficient condition for CQs to be strongly minimal.

EXAMPLE 4.6. *For an example of a strongly minimal CQ, consider query $\mathcal{Q}_1$,*

$$T(x_1,x_2,x_2,x_4) \leftarrow R(x_1,x_2), R(x_2,x_3), R(x_3,x_4).$$

Notice that, by fullness of $\mathcal{Q}_1$, there are no two distinct valuations for $\mathcal{Q}_1$ that derive the same fact. Hence, every valuation of $\mathcal{Q}_1$ must indeed be minimal.

For another example, consider the query $\mathcal{Q}_2$,

$$T() \leftarrow R_1(x_1,x_2), R_2(x_2,x_3), R_3(x_3,x_4).$$

As each atom in the body of $\mathcal{Q}_2$ has a different relation symbol, each valuation of $\mathcal{Q}_2$ yields exactly three different facts and therefore, each valuation is minimal. $\square$

It is easy to see that every strongly minimal CQ is also a minimal CQ, but the converse is not true as witnessed by the query of Example 3.2, which is minimal but not strongly minimal.

The following lemma now provides a characterization of parallel-correctness transfer for strongly minimal queries.

LEMMA 4.7. *Let $\mathcal{Q}'$ be a CQ and let $\mathcal{Q}$ be a strongly minimal CQ. Parallel-correctness transfers from $\mathcal{Q}$ to $\mathcal{Q}'$ if and only if the following holds:*

(C3) there is a simplification θ for $\mathcal{Q}'$ and a substitution ρ for $\mathcal{Q}$ such that $\mathit{body}_{\theta(\mathcal{Q}')} \subseteq \mathit{body}_{\rho(\mathcal{Q})}$.

PROOF. We show that, for strongly minimal $\mathcal{Q}$, (C2) and (C3) are equivalent.

We first show that (C3) implies (C2). It suffices to show that if (C3) holds then for every minimal valuation V' for $\mathcal{Q}'$, there is a valuation V for $\mathcal{Q}$ such that $V'(\mathit{body}_{\mathcal{Q}'}) \subseteq V(\mathit{body}_{\mathcal{Q}})$. By strong minimality of Q, we can then conclude that V is actually minimal.

Let V' be a minimal valuation for $\mathcal{Q}'$ and let θ and ρ be as in (C3). As θ is a simplification, $\mathit{head}_{\theta(\mathcal{Q}')} = \mathit{head}_{\mathcal{Q}'}$ and $\mathit{body}_{\theta(\mathcal{Q}')} \subseteq \mathit{body}_{\mathcal{Q}'}$. Therefore $(V' \circ \theta)$ is also a valuation for $\mathcal{Q}'$ with $(V' \circ \theta)(\mathit{body}_{\mathcal{Q}'}) = V'(\mathit{body}_{\theta(\mathcal{Q}')}) \subseteq V'(\mathit{body}_{\mathcal{Q}'})$ and by minimality of V' the latter inclusion is actually an equality.

By (C3), $\mathit{body}_{\theta(\mathcal{Q}')} \subseteq \mathit{body}_{\rho(\mathcal{Q})}$, therefore V' is a partial valuation for $\rho(\mathcal{Q})$. Let V'' be some arbitrarily chosen extension of V' that is a (total) valuation for $\rho(\mathcal{Q})$. Then,

$$\begin{aligned} V'(\mathit{body}_{\mathcal{Q}'}) = V'(\mathit{body}_{\theta(\mathcal{Q}')}) &= V''(\mathit{body}_{\theta(\mathcal{Q}')}) \\ &\subseteq V''(\mathit{body}_{\rho(\mathcal{Q})}) = (V'' \circ \rho)(\mathit{body}_{\mathcal{Q}}). \end{aligned}$$

Thus, $V \stackrel{\text{def}}{=} V'' \circ \rho$ is the desired valuation for $\mathcal{Q}$.

We next show that (C2) implies (C3). Actually, this implication even holds without the assumption that $\mathcal{Q}$ is strongly minimal. Let us therefore assume that (C2) holds. We choose θ as an arbitrary simplification that minimizes $\mathcal{Q}'$. Such a simplification can be found thanks to [7]. In particular, $\theta(\mathcal{Q}')$ is a minimal CQ that is equivalent to $\mathcal{Q}'$.

Let V' be an injective valuation for $\mathcal{Q}'$. We claim that $V' \circ \theta$ is a minimal valuation for $\mathcal{Q}'$. Towards a contradiction, let us assume that there is a valuation V'' such that $V'' <_{\mathcal{Q}'} V' \circ \theta$. Since θ is the identity on the head variables, V' is injective, and V' and V'' agree on $\mathit{head}_{\mathcal{Q}}$, we can conclude that $((V')^{-1} \circ V'')(\mathit{head}_{\mathcal{Q}'}) = \mathit{head}_{\mathcal{Q}'}$, thus $(V')^{-1} \circ V''$ is a homomorphism from $\mathcal{Q}$ to $((V')^{-1} \circ V'')(\mathcal{Q})$. Furthermore, $((V')^{-1} \circ V'')(\mathit{body}_{\mathcal{Q}}) \subseteq \mathit{body}_{\theta(\mathcal{Q})} \subseteq \mathit{body}_{\mathcal{Q}}$, therefore the identity is a homomorphism from $((V')^{-1} \circ V'')(\mathcal{Q})$ to $\mathcal{Q}$. Together, $((V')^{-1} \circ V'')(\mathcal{Q})$ is equivalent to $\mathcal{Q}$. Furthermore, $((V')^{-1} \circ V'')(\mathit{body}_{\mathcal{Q}}) \subsetneq (V')^{-1}(V'(\mathit{body}_{\mathcal{Q}})) = \mathit{body}_{\theta(\mathcal{Q})}$, contradicting the minimality of θ. We thus conclude that $V' \circ \theta$ is indeed a minimal valuation for $\mathcal{Q}'$.

By (C2), there exists a minimal valuation V for $\mathcal{Q}$ such that $V'(\mathit{body}_{\theta(\mathcal{Q}')}) = (V' \circ \theta)(\mathit{body}_{\mathcal{Q}'}) \subseteq V(\mathit{body}_{\mathcal{Q}})$. Now, let f be an extension of $(V')^{-1}$, which maps values that occur in $V(\mathit{body}_{\mathcal{Q}})$ but not in $V'(\mathit{body}_{\mathcal{Q}'})$ in an arbitrary fashion and let $\rho \stackrel{\text{def}}{=} (f \circ V)$. Then,

$$\begin{aligned} \mathit{body}_{\theta(\mathcal{Q}')} &= (V')^{-1}(V'(\mathit{body}_{\theta(\mathcal{Q}')})) \\ &= f(V'(\mathit{body}_{\theta(\mathcal{Q}')})) = f((V' \circ \theta)(\mathit{body}_{\mathcal{Q}'})) \\ &\subseteq f(V(\mathit{body}_{\mathcal{Q}})) = \rho(\mathit{body}_{\mathcal{Q}}) = \mathit{body}_{\rho(\mathcal{Q})}. \end{aligned}$$

Thus, θ and ρ witness condition (C3). $\square$

THEOREM 4.8. PC-TRANS *restricted to inputs with strongly minimal $\mathcal{Q}$ is* NP*-complete.*

PROOF (SKETCH). The upper bound follows from Lemma 4.7 by the observation that condition (C3) can be checked by a straighforward NP-algorithm. The lower bound follows from Proposition 5.3 below. $\square$

Theorem 4.8 assumes that it is known that Q is strongly minimal. We complete the picture by investigating the complexity of the problem to decide whether a CQ is strongly minimal.

We first give a lemma that generalizes the above examples into a sufficient (but not necessary) condition for strong minimality. In particular, Lemma 4.9 implies that every full CQ and every CQ without self-joins is strongly minimal. We say that an atom in a CQ is a *self-join atom* when the relation name of that atom occurs more than once in $\mathcal{Q}$. For instance, in the query $T() \leftarrow R(x_1, x_2), R(x_2, x_1)$ both $R(x_1, x_2)$ and $R(x_2, x_1)$ are self-join atoms.

LEMMA 4.9. *Let $\mathcal{Q}$ be a CQ. Then $\mathcal{Q}$ is strongly minimal when the following condition holds: if a variable x occurs at a position i in some self-join atom and not in the head of $\mathcal{Q}$, then all self-join atoms have x at position i.*

PROOF (SKETCH). The proof is by contraposition, i.e., we show that if there is a valuation for $\mathcal{Q}$ which is not minimal then the condition is not satisfied. To this end, let V and V' be valuations for $\mathcal{Q}$ which agree on the head-variables and where $V'(body_{\mathcal{Q}}) \subsetneq V(body_{\mathcal{Q}})$.

Then, there are at least two atoms $A_1 = R(x_1, \ldots, x_k)$ and $A_2 = R(y_1, \ldots, y_k)$ in the body of $\mathcal{Q}$ that collapse under V', but not under V. That is, $V'(A_1) = V'(A_2)$ and $V(A_1) \neq V(A_2)$. So, under V' all the variables in A_1 and A_2 on matching positions must be mapped on the same constant, $V'(x_i) = V'(y_i)$ for each $i \in \{1, \ldots, k\}$, while for V there is a position $j \in \{1, \ldots, k\}$ where this is not the case, $V(x_j) \neq V(y_j)$. Obviously, at least one of these variables must then be a non-head variable. So, either only x_j is a head variable, or only y_j is a head variable, or both are distinct non-head variables. In both cases the condition is not satisfied. □

EXAMPLE 4.10. *For an example of a strongly minimal CQ that does not satisfy the condition in Lemma 4.9, consider query $\mathcal{Q}_3$,*

$$T() \leftarrow R(x_1, x_2), R(x_2, x_1).$$

Notice that $\mathcal{Q}_3$ is indeed strongly minimal, because every valuation for $\mathcal{Q}_3$ either maps x_1 and x_2 on the same value, and thus requires only one fact where both values are equal, or maps x_1 and x_2 onto two distinct values, and thus requires exactly two facts where both values are distinct.

Finally, we establish the complexity of deciding strong minimality.

LEMMA 4.11. *Deciding whether a CQ is strongly minimal is* coNP*-complete.*

PROOF (SKETCH). The complement problem is easily seen to be in NP: for two guessed valuations V^*, V (encoded in length polynomial of the query $\mathcal{Q}$) it can be checked in polynomial time whether $V^* <_{\mathcal{Q}} V$.

A lower bound for the complement problem can be obtained via a reduction from 3SAT. □

5. FAMILIES OF DISTRIBUTION POLICIES

Parallel-correctness transfer can be seen as a generalization of parallel-correctness. In both cases, the goal is to decide whether a query can be correctly evaluated by evaluating it locally at each node. However, for parallel-correctness *transfer*, the question whether $\mathcal{Q}'$ is parallel-correct is not asked for a particular distribution policy but for the *family* of those distribution policies, for which $\mathcal{Q}$ is parallel-correct.[8]

In this section, we study the parallel-correctness problem for other kinds of families of distribution policies that can be associated with a given query $\mathcal{Q}$. In Section 5.1, we will identify classes of families of policies, for which (C3) characterizes parallel-correctness. For these classes we conclude that it is NP-complete to decide, whether for the family $\mathcal{F}$ of policies associated with some given CQ $\mathcal{Q}$, a CQ $\mathcal{Q}'$ is parallel-correct for all distributions from $\mathcal{F}$. In Section 5.2, we will see that this, in particular, holds for the families of distribution policies related to the practical Hypercube algorithm, that was previously investigated in several works [3, 5, 6, 8, 9]. In fact, we even show that this holds for a more general class of distribution policies specified in a declarative formalism.

5.1 Parallel-correctness

We start with the following definition:

Definition 7. A query $\mathcal{Q}$ is *parallel-correct for a family $\mathcal{F}$* of distribution policies if it is parallel-correct under every distribution policy from $\mathcal{F}$.

We call a distribution policy $\boldsymbol{P}$ *$\mathcal{Q}$-generous* for a CQ $\mathcal{Q}$, if, for every valuation V for $\mathcal{Q}$, there is a node κ that contains all facts from $V(body_{\mathcal{Q}})$. A family of distribution policies $\mathcal{F}$ is $\mathcal{Q}$-generous if every policy in $\mathcal{F}$ is. For an instance I, a distribution policy $\boldsymbol{P}$ is called *$(\mathcal{Q}, I)$-scattered* if for each node κ there is a valuation V for $\mathcal{Q}$, such that $dist_{\boldsymbol{P},I}(\kappa) \subseteq V(body_{\mathcal{Q}})$. We then say that a family $\mathcal{F}$ of distribution policies is $\mathcal{Q}$-scattered if $\mathcal{F}$ contains a $(\mathcal{Q}, I)$-scattered policy for every I. A $(\mathcal{Q}, I)$-scattered policy that is also $\mathcal{Q}$-generous yields the finest possible partition of the facts of I and thus, intuitively, scatters them as much as possible.

LEMMA 5.1. *Let $\mathcal{Q}$ be a CQ and let $\mathcal{F}$ be a family of distribution policies that is $\mathcal{Q}$-generous and $\mathcal{Q}$-scattered. Then for every CQ $\mathcal{Q}'$, $\mathcal{Q}'$ is parallel correct for $\mathcal{F}$ if and only if:*

(C3) there is a simplification θ for $\mathcal{Q}'$ and a substitution ρ for $\mathcal{Q}$ such that $body_{\theta(\mathcal{Q}')} \subseteq body_{\rho(\mathcal{Q})}$.

We emphasize that Lemma 5.1 uses the same condition (C3) as Lemma 4.7.

PROOF (SKETCH). (if) Let I be a database for $\mathcal{Q}'$, $\boldsymbol{P}$ a distribution policy from $\mathcal{F}$, and let θ and ρ be as guaranteed by (C3). We show that each fact from $\mathcal{Q}'(I)$ is produced at some node. Let V' be a valuation that yields some fact $\boldsymbol{h} \stackrel{\text{def}}{=} V'(head_{\mathcal{Q}'})$ and let V'' be an arbitrary extension of V' for $\rho(\mathcal{Q})$. As θ is a simplification, $(V' \circ \theta)$ also yields the fact $\boldsymbol{h}$. By (C3) we get $(V' \circ \theta)(body_{\mathcal{Q}'}) = V'(body_{\theta(\mathcal{Q}')}) \subseteq V''(body_{\rho(\mathcal{Q})}) = (V'' \circ \rho)(body_{\mathcal{Q}})$. As $\boldsymbol{P}$ is $\mathcal{Q}$-generous, there is some node κ that has all facts from $(V'' \circ \rho)(body_{\mathcal{Q}})$ and therefore all facts from $(V' \circ \theta)(body_{\mathcal{Q}'})$, and thus $\boldsymbol{h}$ is produced at κ.

(only-if) Suppose $\mathcal{Q}'$ is parallel-correct under all distribution policies in $\mathcal{F}$. Let V' be some injective valuation for $\mathcal{Q}'$. Denote $I \stackrel{\text{def}}{=} V'(body_{\mathcal{Q}'})$ and $\boldsymbol{h} \stackrel{\text{def}}{=} V'(head_{\mathcal{Q}'})$. Let $\boldsymbol{P}$ be

[8]A family of distribution policies is just a set of distribution policies.

some $(\mathcal{Q}, I)$-scattered distribution policy from $\mathcal{F}$. Because $\mathcal{Q}'$ is parallel-correct under $\boldsymbol{P}$, there must be a node κ that outputs $\boldsymbol{h}$ when I is distributed according to $\boldsymbol{P}$. Therefore, there is a valuation W' for $\mathcal{Q}'$ such that κ contains all facts from $W'(body_{\mathcal{Q}'})$ and $W'(head_{\mathcal{Q}'}) = \boldsymbol{h}$. We claim that $\theta \stackrel{\text{def}}{=} (V')^{-1} \circ W'$ is a simplification of $\mathcal{Q}'$. Indeed, this substitution is well-defined thanks to the injectivity of V' and furtermore $((V')^{-1} \circ W')(head_{\mathcal{Q}'}) = head_{\mathcal{Q}'}$ and $((V')^{-1} \circ W')(body_{\mathcal{Q}'}) \subseteq body_{\mathcal{Q}'}$, as $W'(body_{\mathcal{Q}'}) \subseteq I = V'(body_{\mathcal{Q}'})$ and $(V')^{-1}$ maps I back to $body_{\mathcal{Q}'}$.

As $\boldsymbol{P}$ is $(\mathcal{Q}, I)$-scattered, there is a valuation V such that $dist_{\boldsymbol{P},I}(\kappa) \subseteq V(body_{\mathcal{Q}})$. Then, let g be some mapping from $img(V)$ to **var** such that for all $d \in img(W')$, $g(d) = g'(d)$. We define the renaming $\rho \stackrel{\text{def}}{=} g \circ V$ and show that with these choices, $body_{\theta(\mathcal{Q}')} \subseteq body_{\rho(\mathcal{Q})}$, and thus (C3) holds.

Let $R(x_1, \ldots, x_k) \in body_{\theta(\mathcal{Q}')}$. Then, there is an atom $R(y_1, \ldots, y_k) \in body_{\mathcal{Q}'}$ with $W'(R(\bar{y})) \in dist_{\boldsymbol{P},I}(\kappa)$ and, for each i, $x_i = (V')^{-1}(W'(y_i))$. So, as $dist_{\boldsymbol{P},I}(\kappa) \subseteq V(body_{\mathcal{Q}})$, $W'(R(\bar{y})) \in V(body_{\mathcal{Q}})$ and there is an atom $R(z_1, \ldots, z_k) \in body_{\mathcal{Q}}$ such that $W'(R(\bar{y})) = V(R(\bar{z}))$. Clearly, $W'(y_i) = V(z_i)$ for all i. By definition of g, it then follows that $x_i = (V')^{-1}(W'(y_i)) = g(V(z_i))$, for all i. Thus, $R(x_1, \ldots, x_k)$ is in $body_{\rho(\mathcal{Q})}$, as desired. $\square$

THEOREM 5.2. *It is* NP*-complete to decide, for given CQs $\mathcal{Q}$ and $\mathcal{Q}'$, whether $\mathcal{Q}'$ is parallel-correct for $\mathcal{Q}$-generous and $\mathcal{Q}$-scattered families of distribution policies.*

The proof of this theorem shows in particular, that $\mathcal{Q}'$ is either parallel-correct for *all* $\mathcal{Q}$-generous and $\mathcal{Q}$-scattered families of distribution policies or for *none* of them.

PROOF (SKETCH). The upper bound follows immediately from Lemma 5.1 and the fact that (C3) can be checked by an NP-algorithm. Indeed such an algorithm only needs to guess θ and ρ and to verify (in polynomial time) that $body_{\theta(\mathcal{Q}')} \subseteq body_{\rho(\mathcal{Q})}$.

The lower bound follows by Lemma 5.1 and the following Proposition 5.3. $\square$

PROPOSITION 5.3. *It is* NP*-hard to decide, whether for CQs $\mathcal{Q}$ and $\mathcal{Q}'$ condition (C3) holds. This statement remains true if either $\mathcal{Q}$ or $\mathcal{Q}'$ is restricted to acyclic queries. It also remains true if both CQs are Boolean and if $\mathcal{Q}$ is full.*

REMARK 5.4. *The proof of Proposition 5.3 in both cases ($\mathcal{Q}$ acyclic or $\mathcal{Q}'$ acyclic) is by a reduction from graph* 3*-colorability. The first reduction, in which the input graph is encoded in $\mathcal{Q}'$ and the valid color-assignments in $\mathcal{Q}$ is straightforward. As it only uses a fixed number of colors, $\mathcal{Q}$ can be made acyclic by adding an atom to $\mathcal{Q}$ that contains all allowed colors.*

The second reduction, in which the graph is encoded in $\mathcal{Q}$ and the valid color-assignments in $\mathcal{Q}'$, is a bit more involved.

The reader may now wonder whether NP*-hardness remains when both $\mathcal{Q}$ and $\mathcal{Q}'$ are required to be acyclic. When relations of arbitrary arity are allowed, this is indeed the case: acyclicity is then easily achieved by using one atom containing all variables of the query. Under bounded-arity database schemas, however, the complexity of parallel-correctness transfer for acylic queries remains open.* $\square$

5.2 Hypercube Distribution Policies

In the following, we give a short definition of Hypercube distributions and settle the complexity of the parallel-correctness transfer problem for families $\mathcal{H}(\mathcal{Q})$ of Hypercube distributions for some CQ $\mathcal{Q}$ with the help of the results of Section 5.1. We highlight how Hypercube distributions can be specified in a rule-based fashion, which we consider useful also for more general distributions.

Let $\mathcal{Q}$ be a conjunctive query with variables $x_1, \ldots, x_k$. A collection $H = (h_1, \ldots, h_k)$ of hash functions[9] (called a *hypercube* in the following) determines a *hypercube distribution $\boldsymbol{P}_H$* for $\mathcal{Q}$ in the following way. For each $i \in \{1, \ldots, k\}$, we let $A_i \stackrel{\text{def}}{=} img(h_i)$ and define the *address space $\mathcal{A}$* of $\boldsymbol{P}_H$ as the cartesian product $A_1 \times \cdots \times A_k$.

In a nutshell, $\boldsymbol{P}_H$ has one node per address in $\mathcal{A}$ and distributes, for every valuation V of $\mathcal{Q}$, every fact $\boldsymbol{f} = V(A)$, where A is an atom of $\mathcal{Q}$, to all nodes whose address $(a_1, \ldots, a_k)$ satisfies $a_i = h_i(V(x_i))$, for all variables x_i occurring in A.

For the declarative specification of $\boldsymbol{P}_H$ we make use of predicates[10] $bucket_i$ and $bucket^*_i$, where $bucket_i(a, b)$ holds, if $h_i(a) = b$, and $bucket^*_i(b)$ holds, if $b \in img(h_i)$.

With these predicates, $\boldsymbol{P}_H$ can be specified by stating, for each atom $R(y_1, \ldots, y_m)$ of $\mathcal{Q}$, one rule

$$\begin{aligned} T_R(z_1, \ldots, z_k; y_1, \ldots, y_m) \leftarrow & R(y_1, \ldots, y_m), \\ & B_1, \ldots, B_k. \end{aligned}$$

Here, for each $i \in \{1, \ldots, k\}$, B_i is $bucket_i(x_i, z_i)$, if x_i occurs in $y_1, \ldots, y_m$, and B_i is $bucket^*_i(z_i)$, otherwise.

The semantics of such a rule is straightforward. For each valuation V of the variables $z_1, \ldots, z_k, x_1, \ldots, x_k$, that makes the body of the rule true, the fact $R(V(y_1), \ldots, V(y_m))$ is distributed to the node with address $(V(z_1), \ldots, V(z_k))$. We emphasize that the variables $y_1, \ldots, y_m$ need not be pairwise distinct and that $\{y_1, \ldots, y_m\} \subseteq \{x_1, \ldots, x_k\}$.

REMARK 5.5. *It is evident that one could use more general rules to specify distribution policies. More than one atom with a database relation could be in the body, and there could be other additional predicates than those derived from hashing functions. Furthermore, the address space could be defined differently.* $\square$

For a CQ $\mathcal{Q}$, we denote by $\mathcal{H}_{\mathcal{Q}}$ the family of distribution policies $\{\boldsymbol{P}_H \mid H \text{ is a hypercube for } \mathcal{Q}\}$.

LEMMA 5.6. *Let $\mathcal{Q}$ be a CQ. Then $\mathcal{H}_{\mathcal{Q}}$ is $\mathcal{Q}$-generous and $\mathcal{Q}$-scattered.*

PROOF. Let $\mathcal{Q}$ be a CQ with $vars(\mathcal{Q}) = \{u_1, \ldots, u_k\}$.

We first show that every policy $\boldsymbol{P}_H \in \mathcal{H}_{\mathcal{Q}}$ is $\mathcal{Q}$-generous. To this end, let H be a hypercube and let V be a valuation for $\mathcal{Q}$. Then, by definition, for the node κ with address $(h_1(V(u_1)), \ldots, h_k(V(u_k)))$, $\kappa \in \boldsymbol{P}_H(\boldsymbol{f})$ for every $\boldsymbol{f} \in V(body_{\mathcal{Q}})$.

We now show that $\mathcal{H}_{\mathcal{Q}}$ is $\mathcal{Q}$-scattered. Thereto, let I be an instance. For every $i \leq k$, we choose $A_i \stackrel{\text{def}}{=} adom(I)$ and let $h_i(a) \stackrel{\text{def}}{=} a$, for every $a \in A_i$. Let κ be an arbitrary node and

[9] A hash function is a partial mapping from **dom** to a finite set whose elements are sometimes referred to as buckets.

[10] For the purpose of specification it is irrelevant whether these predicates are materialized in the database.

let $(a_1, \ldots, a_k)$ be its address. Let V be the valuation mapping u_i to a_i, for each i. Let $R(d_1, \ldots, d_m) \in \mathit{dist}_{\boldsymbol{P}_H, I}(\kappa)$ thanks to some rule

$$T_R(z_1, \ldots, z_k; y_1, \ldots, y_m) \leftarrow R(y_1, \ldots, y_m), \\ B_1, \ldots, B_k.$$

By definition of the hash functions, every valuation that satisfies the body of this rule, maps x_i to a_i, for every x_i that appears in $R(y_1, \ldots, y_m)$. However, as this valuation coincides with V on $y_1, \ldots, y_m$, it maps $R(y_1, \ldots, y_m)$ to an element of $V(\mathit{body}_{\mathcal{Q}})$. Therefore, $\mathit{dist}_{\boldsymbol{P}_H, I}(\kappa) \subseteq V(\mathit{body}_{\mathcal{Q}})$. $\square$

COROLLARY 5.7. *It is* NP*-complete to decide, for given conjunctive queries $\mathcal{Q}, \mathcal{Q}'$, whether $\mathcal{Q}'$ is parallel-correct for $\mathcal{H}_{\mathcal{Q}}$.*

REMARK 5.8. *It is easy to see that Lemma 5.6 and then the upper bound of Corollary 5.7 holds for more general families of distribution policies. As an example, one could add further atoms of $\mathcal{Q}$ as "filters" to the bodies of the above rules.* $\square$

6. CONCLUSIONS

We have introduced parallel-correctness as a framework for studying one-round evaluation algorithms for the evaluation of queries under arbitrary distribution policies. We have obtained tight bounds on the complexity of deciding parallel-correctness and the transferability problem for conjunctive queries. For general conjunctive queries, these complexities reside in different levels of the polynomial hierarchy (even when considering Hypercube distributions). Since the considered problems are static analysis problems that relate to queries and not to instances (at least in the case of transferability), such complexities do not necessarily put a burden on practical applicability. Still, it would be interesting to identify fragments of conjunctive queries or particular classes of distribution policies that could render these problems tractable. In addition, it would be interesting to explore more expressive classes of queries like unions of CQs and CQs with negation, and other families of distribution policies.

The notion of parallel-correctness is directly inspired by Hypercube where the result of the query is obtained by aggregating (through union) the evaluation of the original query over the distributed instance. Other possibilities are to consider more complex aggregator functions than union and to allow for a different query than the original one to be executed at computing nodes.

Acknowledgments

We thank Serge Abiteboul, Luc Segoufin, Cristina Sirangelo, and Thomas Zeume for helpful remarks.

7. REFERENCES

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

[2] F. N. Afrati, P. Koutris, D. Suciu, and J. D. Ullman. Parallel skyline queries. In *ICDT*, pages 274–284, 2012.

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

[4] T. J. Ameloot, B. Ketsman, F. Neven, and D. Zinn. Weaker forms of monotonicity for declarative networking: a more fine-grained answer to the CALM-conjecture. In *PODS*, pages 64–75, 2014.

[5] P. Beame, P. Koutris, and D. Suciu. Communication steps for parallel query processing. In *PODS*, pages 273–284, 2013.

[6] P. Beame, P. Koutris, and D. Suciu. Skew in parallel query processing. In *PODS*, pages 64–75, 2014.

[7] A. K. Chandra and P. M. Merlin. Optimal implementation of conjunctive queries in relational data bases. In *STOC*, pages 77–90, 1977.

[8] S. Ganguly, A. Silberschatz, and S. Tsur. A framework for the parallel processing of Datalog queries. In *SIGMOD*, pages 143–152, 1990.

[9] S. Ganguly, A. Silberschatz, and S. Tsur. Parallel bottom-up processing of datalog queries. *J. Log. Program.*, 14(1&2):101–126, 1992.

[10] P. Hell and J. Nesetril. The core of a graph. *Discrete Mathematics*, 109(1-3):117–126, 1992.

[11] P. Koutris and D. Suciu. Parallel evaluation of conjunctive queries. In *PODS*, pages 223–234, 2011.

[12] Spark. http://spark.apache.org.

[13] L. J. Stockmeyer. The polynomial-time hierarchy. *Theor. Comput. Sci.*, 3(1):1–22, 1976.

[14] R. Xin, J. Rosen, M. Zaharia, M. Franklin, S. Shenker, and I. Stoica. Shark: SQL and rich analytics at scale. In *SIGMOD*, 2013.

[15] D. Zinn, T. J. Green, and B. Ludäscher. Win-move is coordination-free (sometimes). In *ICDT*, pages 274–284, 2012.