The Homomorphism Problem for Regular Graph Patterns

Miguel Romero
Simons Inst. for the Theory of Comp.
Berkeley, University of California
m.romero.orth@gmail.com

Pablo Barceló
Center for Semantic Web Research &
DCC, University of Chile
pbarcelo@dcc.uchile.cl

Moshe Y. Vardi
Department of Computer Science
Rice University
vardi@cs.rice.edu

***Abstract*—The evaluation of conjunctive regular path queries – which form the navigational core of the query languages for graph databases – raises challenges in the context of the homomorphism problem that are not fully addressed by existing techniques. We start a systematic investigation of such challenges using a notion of homomorphism for regular graph patterns (RGPs). We observe that the RGP homomorphism problem cannot be reduced to known instances of the homomorphism problem, and new techniques need to be developed for its study.**

We first show that the non-uniform version of the problem is computationally harder than for the usual homomorphism problem. By establishing a connection between both problems, in turn, we postulate a dichotomy conjecture, analogous to the algebraic dichotomy conjecture held in CSP. We also look at which structural restrictions on left-hand side instances of the RGP homomorphism problem ensure efficiency. We study restrictions based on the notion of bounded treewidth modulo equivalence, which characterizes tractability for the usual homomorphism notion. We propose two such notions, based on different interpretations of RGP equivalence, and show that they both ensure the efficiency of the RGP homomorphism problem.

I. Introduction

The homomorphism problem. The *homomorphism problem for relational structures* – given relational structures $\mathcal{A}$ and $\mathcal{B}$, is there a homomorphism $h : \mathcal{A} \to \mathcal{B}$? – provides an equivalent reformulation of two fundamental tasks in computer science: (a) the constraint satisfaction problem (CSP) [1], and (b) conjunctive query (CQ) evaluation in databases [2]. In general, this problem is NP-complete. This has motivated a long line of work whose main goal is to understand which restrictions of the problem are tractable. Such question has been studied mainly from two points of view, described next:

The non-uniform homomorphism problem: Many CSPs of interest can be reformulated as a *non-uniform* homomorphism problem in which the *template* $\mathcal{B}$ is fixed and the input consists of the *instance* $\mathcal{A}$ only; written as $\mathsf{Hom}(\mathcal{B})$. The Dichotomy Conjecture of Feder and Vardi postulates that $\mathsf{Hom}(\mathcal{B})$ is either tractable or NP-complete [3]. A series of groundbreaking results have given rise to a refined version of this conjecture – the so-called Algebraic Dichotomy Conjecture (ADC) – which postulates that $\mathsf{Hom}(\mathcal{B})$ is tractable if $\mathcal{B}$ has certain algebraic property; otherwise, it is NP-complete. Only the NP-completeness side of the ADC is known to hold [4], [5], [6].[1]

Structural restrictions on the left-hand side: This is of interest in the context of CQ evaluation over relational databases, i.e., the problem of checking if $\mathcal{B} \models q$, where (i) q is a (Boolean) CQ, i.e., a sentence in the $\{\exists, \wedge\}$-fragment of FO of the form $\exists \bar{y} \bigwedge_{1 \leq i \leq m} R_i(\bar{x}_i)$, where the $R_i(\bar{x}_i)$'s are atoms, and (ii) $\mathcal{B}$ is a relational structure (the database). This problem is equivalent to checking whether $\mathcal{A} \to \mathcal{B}$, where $\mathcal{A} = \{R_i(\bar{x}_i) \mid 1 \leq i \leq m\}$. The input now consists of both $\mathcal{A}$ and $\mathcal{B}$, and we are interested in identifying which classes $\mathbb{C}$ of $\mathcal{A}$'s (resp., CQs) ensure tractability of the homomorphism problem (resp., CQ evaluation). Formally, we want to understand when $\mathsf{Hom}(\mathbb{C}, -)$ – the restriction of the homomorphism problem in which the structure $\mathcal{A}$ is in $\mathbb{C}$ – is tractable. A prime example of this is when the class $\mathbb{C}$ is of *bounded treewidth* [10], [11]. Such good behavior also extends to the $\mathbb{C}$'s that are of bounded treewidth *modulo equivalence*, i.e., those for which there is an integer $k \geq 1$ such that every $\mathcal{A}$ in $\mathbb{C}$ is homomorphically equivalent to an $\mathcal{A}'$ of treewidth k [11]. As shown by Grohe [12], for schemas of fixed arity this notion exhausts the space of classes $\mathbb{C}$ for which $\mathsf{Hom}(\mathbb{C}, -)$ is tractable (under complexity assumptions). That is, $\mathsf{Hom}(\mathbb{C}, -)$ is tractable iff $\mathbb{C}$ has bounded treewidth modulo equivalence.

Many extensions of the homomorphism problem have been studied in the literature. One such an extension that is particularly relevant to us is the one that relates to the evaluation of *existential positive* FO ($\exists^+$FO) formulas. Its non-uniform version (denoted $\exists^+\mathrm{FO}(\mathcal{B})$) has been studied in [13], [14], while the problem of which classes of $\exists^+$FO formulas ensure efficient solvability of evaluation was studied in [15].

Here we study another extension of the homomorphism problem, of high relevance in the context of evaluation of *navigational* queries over the emerging data model of *graph databases*, i.e., edge-labeled directed graphs [16]. This problem relates to the evaluation of queries in the most basic such language: the *conjunctive regular path queries* (CRPQs) [17], that extend CQs with the ability to check for the existence of a path that satisfies a regular condition. CRPQs form the core of practical graph query languages such as SPARQL 1.1

[1] Different proofs of the ADC have recently been announced in three articles [7], [8], [9], but such proofs have not been peer reviewed yet.

978-1-5090-3018-7/17/$31.00

[18] and PGQL [19] (cf., [20]), and constitute an active area of theoretical research [21], [22], [23]. Our goal is to develop an in-depth study of the homomorphism problem for CRPQs, and advance the understanding of when such a problem can be efficiently solved. It turns out to be the case that this problem stands on its own right: it cannot be reduced to known instances of the homomorphism problem, and requires developing new techniques to fully realize its potential.

Homomorphisms for regular graph patterns. As customary, we deal not only with CRPQs, but with its *two-way* extension, C2RPQs [24], that allow to traverse edges in both directions. Boolean C2RPQs are sentences of the form:

$$\phi \quad = \quad \exists \bar{z} \left(L_1(x_1, y_1) \wedge \cdots \wedge L_m(x_m, y_m) \right), \qquad (1)$$

where the L_i's are regular expressions over the alphabet of edge labels and their inverses. A CRPQ is simply a C2RPQ that does not use such inverses. In C2RPQ evaluation, we are thus interested in checking if $\mathcal{G} \models \phi$, where $\mathcal{G}$ is a graph database, and ϕ is a C2RPQ of the form (1). As for CQs, this boils down to checking if there is a certain "homomorphism" from ϕ to $\mathcal{G}$. The notion of homomorphism now needed, though, is more flexible than the one used in CQ evaluation. This is because ϕ no longer can be seen as a relational structure, but corresponds to a *two-way regular graph pattern* (2RGP) $\mathcal{P} = \{L_i(x_i, y_i) \mid 1 \leq i \leq m\}$, where the L_i's are regular expressions. A homomorphism from $\mathcal{P}$ to a graph database $\mathcal{G}$ is allowed to map the pair (x_i, y_i) of variables to a pair (u, v) of nodes in $\mathcal{G}$ that is linked by some path labeled in the regular language L_i. This is more general than the standard homomorphism, where each L_i is an atomic edge.[2]

This flexibility poses new demands on the homomorphism problem, not all of which can be solved with its existing toolbox. In consequence, we lack a thorough understanding of the problems $\mathsf{2RGPHom}(\mathcal{G})$ – the non-uniform 2RGP homomorphism problem in which the template is a graph database $\mathcal{G}$ – and $\mathsf{2RGPHom}(\mathbb{C}, -)$ – the 2RGP homomorphism problem in which the 2RGPs $\mathcal{A}$ on the left-hand side are in class $\mathbb{C}$. In particular, we have no answers to the following questions:

$\mathsf{2RGPHom}(\mathcal{G})$: When can we efficiently solve $\mathsf{2RGPHom}(\mathcal{G})$? Can this problem be computationally harder than $\mathsf{Hom}(\mathcal{G})$? How does this relate to the ADC? (Recall that the ADC can be reduced to its version over graph databases [27]).

$\mathsf{2RGPHom}(\mathbb{C}, -)$: For which classes $\mathbb{C}$ can we efficiently solve $\mathsf{2RGPHom}(\mathbb{C}, -)$? In other words, when is C2RPQ evaluation efficiently solvable? How does this relate to known structural restrictions that yield tractability for $\mathsf{Hom}(\mathbb{C}, -)$, e.g., bounded treewidth modulo equivalence?

Our goal is to start a systematic investigation of the 2RGP homomorphism problem. In particular, we want to gain understanding of when $\mathsf{2RGPHom}(\mathcal{G})$ is tractable and which structural restrictions on $\mathbb{C}$ ensure efficiency for $\mathsf{2RGPHom}(\mathbb{C}, -)$.

Our results. We divide our contributions as follows:

[2]The 2RGP homomorphism problem is different to the *subgraph homeomorphism problem* studied in [25], [26]: the latter is defined only for directed graphs, and maps edges of $\mathcal{P}$ to pairwise node-disjoint simple paths in $\mathcal{G}$.

The non-uniform 2RGP homomorphism problem: C2RPQs extend CQs over graph databases. Hence, if $\mathsf{2RGPHom}(\mathcal{G})$ is tractable then so is $\mathsf{Hom}(\mathcal{G})$. We observe that the converse is not true: there is a graph database $\mathcal{G}$ such that $\mathsf{Hom}(\mathcal{G})$ is tractable but $\mathsf{2RGPHom}(\mathcal{G})$ is NP-complete; i.e., the class of $\mathcal{G}$'s for which $\mathsf{2RGPHom}(\mathcal{G})$ is tractable is strictly contained in the class for which $\mathsf{Hom}(\mathcal{G})$ is tractable (assuming $\mathrm{P} \neq \mathrm{NP}$). Moreover, all this holds even in the absence of inverses. Thus, we cannot simply apply the ADC in our context, and we need to develop new tools to study our problem.

We first deal with the case in which no inverses are allowed, i.e., we study $\mathsf{RGPHom}(\mathcal{G})$, the problem of evaluating RGPs without inverses over $\mathcal{G}$. In such context we establish a connection between the RGP and the usual homomorphism problem, as follows: for each graph database $\mathcal{G}$ there is a relational structure $\mathcal{E}_\mathcal{G}$ such that $\mathsf{RGPHom}(\mathcal{G})$ and $\mathsf{Hom}(\mathcal{E}_\mathcal{G})$ are polynomially interreducible. Thus, $\mathsf{RGPHom}(\mathcal{G})$ is tractable (resp., NP-complete) if $\mathsf{Hom}(\mathcal{E}_\mathcal{G})$ is tractable (resp., NP-complete). Moreover, if a dichotomy for the $\mathsf{Hom}(\mathcal{G})$'s holds, then a dichotomy for the $\mathsf{RGPHom}(\mathcal{G})$'s also holds. (Notice that the converse is not implied since a dichotomy for the $\mathcal{E}_\mathcal{G}$'s does not directly imply a dichotomy for *all* templates).

The aforementioned connection also allows us to postulate a conjecture for the non-uniform problem $\mathsf{RGPHom}(\mathcal{G})$ – analogous to the ADC – based on an algebraic property over $\mathcal{G}$. As for ADC, which uses a notion of *polymorphism* that is connected to *CQ-definability*, our conjecture uses a notion of polymorphism that characterizes *CRPQ-definability* [28].

Adding inverses, on the other hand, turns the problem even computationally harder; that is, there are graph databases $\mathcal{G}$ such that $\mathsf{RGPHom}(\mathcal{G})$ is tractable but $\mathsf{2RGPHom}(\mathcal{G})$ is NP-complete. Moreover, we prove some general intractability results for the problem $\mathsf{2RGPHom}(\mathcal{G})$ that suggest that the conditions that ensure tractability in this case must be quite restrictive. Thus, it is possible that a dichotomy can more easily be obtained for the problem $\mathsf{2RGPHom}(\mathcal{G})$ than for its analogue $\mathsf{RGPHom}(\mathcal{G})$ without inverses. We finally connect $\mathsf{2RGPHom}(\mathcal{G})$ with $\exists^+\mathrm{FO}(\mathcal{G})$ – the evaluation of $\exists^+\mathrm{FO}$ sentences over fixed template $\mathcal{G}$ – and show that the former is polynomially reducible to the latter, but not the opposite.

Structural restrictions: We study which classes $\mathbb{C}$ of 2RGPs ensure the efficiency of $\mathsf{2RGPHom}(\mathbb{C}, -)$. Recall that for relational structures (over fixed arity schemas such as the ones used for representing graph databases) the $\mathbb{C}$'s that ensure tractability of $\mathsf{Hom}(\mathbb{C}, -)$ are those of bounded treewidth modulo (homomorphic) equivalence. We use this result as a yardstick, and study how bounded treewidth modulo equivalence relates to the efficiency of $\mathsf{2RGPHom}(\mathbb{C}, -)$.

The crucial difference is that while the notion of homomorphic equivalence for relational structures is well-defined, it is not for 2RGPs. This is because homomorphisms, as studied in the paper, are defined from a 2RGP $\mathcal{P}$ to a graph database $\mathcal{G}$, but not from $\mathcal{P}$ to another 2RGP $\mathcal{P}'$. Moreover, there seems to be no *canonical* way to define "homomorphic equivalence for 2RGPs". We thus propose two natural notions of homomorphic equivalence for 2RGPs – the first one based

on a suitable class of homomorphisms for 2RGPs and the second one on the logical equivalence of their underlying C2RPQs – and prove that they yield classes of 2RGPs of bounded treewidth modulo equivalence with good properties in terms of the 2RGP homomorphisms. In fact, the first class ensures tractability for $\mathsf{2RGPHom}(\mathbb{C}, -)$, and the second one *fixed-parameter tractability* (recall that this is a desirable property for $\mathsf{2RGPHom}(\mathbb{C}, -)$, not held by all classes of 2RGPs [29]). These classes are also conceptually important: While the one based on homomorphisms for 2RGPs corresponds to a natural extension of the notion of bounded treewidth modulo equivalence for relational structures – and actually allows for the application of similar tools based on the *existential pebble game* [26] –, the one based on C2RPQ equivalence is relevant for the evaluation and optimization of C2RPQs [23].

We finally study a slight extension of the notion of bounded treewidth modulo "C2RPQ equivalence" – based on equivalence for *unions* of C2RPQs – for which the 2RGP homomorphism problem remains fixed-parameter tractable. This is, in a sense, optimal, as we prove that the problem is NP-complete.

Organization. Preliminares are in Section II. The non-uniform problem is studied in Section III and structural restrictions based on bounded treewidth in Section IV. In Section V, we study restrictions based on equivalence for unions of C2RPQs. We finish in Section VI with conclusions and open problems.

II. Preliminaries

Relational structures and homomorphisms. Let Dom be a countably infinite set of elements. A *schema* σ is a set of relation symbols, each one of which has an associated arity $n > 0$. A *fact* over σ is an expression of the form $R(\bar{c})$, where R is a relation symbol in σ of arity $n > 0$ and $\bar{c}$ is an n-tuple over Dom. A *relational structure* $\mathcal{A}$ over σ is a set of facts over σ. We write $\mathsf{Dom}(\mathcal{A})$ for the *domain* of $\mathcal{A}$, i.e., the set of elements from Dom that are mentioned in the facts of $\mathcal{A}$. From now on we assume all relational structures to be finite.

Let $\mathcal{A}$ and $\mathcal{B}$ be relational structures. A *homomorphism* from $\mathcal{A}$ to $\mathcal{B}$ is a mapping $h : \mathsf{Dom}(\mathcal{A}) \to \mathsf{Dom}(\mathcal{B})$ such that, for each fact $R(\bar{c})$ in $\mathcal{A}$, it is the case that $R(h(\bar{c}))$ belongs to $\mathcal{B}$. We write $h : \mathcal{A} \to \mathcal{B}$ to denote that h is a homomorphism from $\mathcal{A}$ to $\mathcal{B}$, and $\mathcal{A} \to \mathcal{B}$ to state that at least one such a homomorphism exists. The *homomorphism problem for relational structures* is defined as follows: Given relational structures $\mathcal{A}$ and $\mathcal{B}$ over σ, is it the case that $\mathcal{A} \to \mathcal{B}$?

CQ evaluation. The problem of Boolean conjunctive query (CQ) evaluation over relational databases is equivalent to the homomorphism problem (see, e.g., [30]). Indeed, recall that the former is the problem of checking if $\mathcal{B} \models q$, given a relational structure $\mathcal{B}$ (the relational database) and a Boolean query q – the CQ – which corresponds to a formula in the $\{\exists, \wedge\}$-fragment of FO of the form $\exists \bar{y}(R_1(\bar{x}_1) \wedge \cdots \wedge R_m(\bar{x}_m))$, where the $R_i(\bar{x}_i)$'s are atomic relational formulas such that $\bar{x}_i$ is a tuple of variables. Given a Boolean CQ $q = \exists \bar{y} \bigwedge_{1 \leq i \leq m} R_i(\bar{x}_i)$, we denote by $\mathcal{A}_q$ the relational structure that is obtained from the set $\{R_i(\bar{x}_i) \mid 1 \leq i \leq m\}$ by replacing each variable x with a fresh element $c \in \mathsf{Dom}$. Analogously, given a relational structure $\mathcal{A} = \{R_1(\bar{c}_1), \ldots, R_m(\bar{c}_m)\}$, we write $q_{\mathcal{A}}$ for the Boolean CQ $\exists \bar{y} \bigwedge_{1 \leq i \leq m} R_i(\bar{x}_i)$, where $\{R_1(\bar{x}_1), \ldots, R_m(\bar{x}_m)\}$ is the set that is obtained from $\mathcal{A}$ by replacing each constant $c \in \mathsf{Dom}(\mathcal{A})$ with a fresh variable x. It is clear then that for each relational structure $\mathcal{B}$:

$$\mathcal{B} \models q \iff \mathcal{A}_q \to \mathcal{B} \quad \text{and} \quad \mathcal{A} \to \mathcal{B} \iff \mathcal{B} \models q_{\mathcal{A}}.$$

C2RPQ evaluation. A *graph database* $\mathcal{G}$ is a finite directed graph whose edges are labeled over a countable alphabet Σ. We represent such $\mathcal{G}$ as a relational structure over the schema $\{P_a \mid a \in \Sigma\}$, where the P_a's are binary relation symbols. Then $\mathsf{Dom}(\mathcal{G})$ corresponds to the set of nodes of the graph database, and a fact of the form $P_a(u, v)$ in $\mathcal{G}$, for $a \in \Sigma$, represents the presence of an a-labeled edge from node u to node v. We often write $u \overset{a}{\rightsquigarrow} v$ for the fact $P_a(u, v)$.

A *path* in the graph database $\mathcal{G}$ is a sequence:

$$\rho = v_0 \overset{a_1}{\rightsquigarrow} v_1 \overset{a_2}{\rightsquigarrow} v_2 \ldots v_{k-1} \overset{a_k}{\rightsquigarrow} v_k,$$

for $k \geq 0$, such that $v_{i-1} \overset{a_i}{\rightsquigarrow} v_i$ is in $\mathcal{G}$ for each $1 \leq i \leq k$. For querying purposes, we are typically interested in the *label* of such path, denoted $\lambda(\rho)$, which is the word $a_1 \ldots a_k \in \Sigma^*$.

Conjunctive two-way regular path queries (C2RPQs) express properties of paths in graph databases that can traverse edges in both directions. A clean way to handle the backward traversal of edges is by using the notion of *completion* of a graph database, defined next. Let Σ be a finite alphabet. We write $\Sigma^{\pm}$ for the alphabet that extends Σ with the *inverse* a^- of each symbol $a \in \Sigma$. Given a graph database $\mathcal{G}$ over Σ, we define its completion $\mathcal{G}^{\pm}$ as the graph database over $\Sigma^{\pm}$ that extends $\mathcal{G}$ with the edge $v \overset{a^-}{\rightsquigarrow} u$, for each edge $u \overset{a}{\rightsquigarrow} v$ in $\mathcal{G}$. For a regular expression L over $\Sigma^{\pm}$, we write $L(\mathcal{G})$ for the set of pairs $(u, v) \in \mathsf{Dom}(\mathcal{G}) \times \mathsf{Dom}(\mathcal{G})$ such that there is a path ρ from u to v in $\mathcal{G}^{\pm}$ for which $\lambda(\rho)$ matches L.

A C2RPQ over Σ is a formula $\phi = \exists \bar{z} \bigwedge_{1 \leq i \leq m} x_i \overset{L_i}{\rightsquigarrow} y_i$, where each L_i is a regular expression over $\Sigma^{\pm}$. A CRPQ is a C2RPQ that does not mention the inverses a^-, for $a \in \Sigma$. For a Boolean C2RPQ ϕ as above and a graph database $\mathcal{G}$, we write $\mathcal{G} \models \phi$ if there is a mapping $h : \bigcup_{1 \leq i \leq m} \{x_i, y_i\} \to \mathsf{Dom}(\mathcal{G})$ such that $(h(x_i), h(y_i)) \in L_i(\mathcal{G})$, for each $1 \leq i \leq m$.

The *Boolean C(2)RPQ evaluation problem* takes as input a graph database $\mathcal{G}$ and a C(2)RPQ ϕ, and asks if $G \models \phi$.

Regular graph patterns and homomorphisms. As in the case of CQs, Boolean C2RPQ evaluation can be recast in terms of the homomorphism problem. To do this, however, it is necessary to introduce more general notions of structure and homomorphism than the ones we have used so far. This is because the structures that represent C2RPQs have to be able to express facts based on regular expressions, and the notion of homomorphism used for C2RPQ evaluation has to be able to give proper semantics to such facts. We do this by using the notion of *regular graph pattern* (cf. [31]), as described next.

A two-way regular graph pattern (2RGP) $\mathcal{P}$ is a graph database over $\mathsf{Reg}(\Sigma^{\pm})$, where $\mathsf{Reg}(\Sigma^{\pm})$ is the set of regular expressions over $\Sigma^{\pm}$. An RGP is a 2RGP that does not mention

the inverses a^-, for $a \in \Sigma$. A homomorphism from 2RGP $\mathcal{P}$ to graph database $\mathcal{G}$ is a mapping $h : \mathsf{Dom}(\mathcal{P}) \to \mathsf{Dom}(\mathcal{G})$ such that for each edge $u \overset{L}{\rightsquigarrow} v$ in $\mathcal{P}$, where $u, v \in \mathsf{Dom}(\mathcal{P})$ and $L \in \mathsf{Reg}(\Sigma^{\pm})$, it is the case that $(h(u), h(v)) \in L(\mathcal{G})$. We write $h : \mathcal{P} \to \mathcal{G}$ if h is a homomorphism from $\mathcal{P}$ to $\mathcal{G}$, and $\mathcal{P} \to \mathcal{G}$ if one such a homomorphism exists.

The *(2)RGP homomorphism problem* is as follows: Given a (2)RGP $\mathcal{P}$ and a graph database $\mathcal{G}$, is it the case that $\mathcal{P} \to \mathcal{G}$? In general, these problems are NP-complete.

C2RPQ evaluation and 2RGP homomorphisms. C2RPQ evaluation can be recast in terms of the 2RGP homomorphism problem. Given a Boolean C(2)RPQ ϕ of the form $\exists \bar{z} \bigwedge_{1 \leq i \leq m} x_i \overset{L_i}{\rightsquigarrow} y_i$ over Σ, we denote by $\mathcal{P}_\phi$ the (2)RGP which is obtained from $\{x_i \overset{L_i}{\rightsquigarrow} y_i \mid 1 \leq i \leq m\}$ by replacing each variable z with a fresh element c in Dom. Analogously, given a (2)RGP $\mathcal{P} = \{u_i \overset{L_i}{\rightsquigarrow} v_i \mid 1 \leq i \leq m\}$, we denote by $\phi_\mathcal{P}$ the Boolean C(2)RPQ $\exists \bar{z} \bigwedge_{1 \leq i \leq m} x_i \overset{L_i}{\rightsquigarrow} y_i$, where $\{x_i \overset{L_i}{\rightsquigarrow} y_i \mid 1 \leq i \leq m\}$ is obtained from $\mathcal{P}$ by replacing each constant $u \in \mathsf{Dom}(\mathcal{P})$ by a fresh variable w. Then:

Proposition II.1. *For each C(2)RPQ ϕ and (2)RGP $\mathcal{P}$:*

$$\mathcal{G} \models \phi \iff \mathcal{P}_\phi \to \mathcal{G} \quad \text{and} \quad \mathcal{P} \to \mathcal{G} \iff \mathcal{G} \models \phi_\mathcal{P},$$

over each graph database $\mathcal{G}$.

Hence, the C(2)RPQ evaluation problem is NP-complete.

III. The non-uniform RGP homomorphism problem

Let $\mathcal{G}$ be a graph database. We define $\mathsf{2RGPHom}(\mathcal{G})$ as the problem of checking if $\mathcal{P} \to \mathcal{G}$, given a 2RGP $\mathcal{P}$. We study for which graph databases $\mathcal{G}$ is $\mathsf{2RGPHom}(\mathcal{G})$ tractable.

A. Connections with the non-uniform homomorphism problem

For each graph database $\mathcal{G}$ the problem $\mathsf{Hom}(\mathcal{G})$ is polynomially reducible to $\mathsf{2RGPHom}(\mathcal{G})$ (since 2RGPs extend graph databases). It is natural to ask if the opposite is also true. We prove next that it is not (under complexity theoretical assumptions). Moreover, this holds even in the absence of inverses. Formally, let us write $\mathsf{Hom}_{\text{NP-c}}$ and $\mathsf{RGPHom}_{\text{NP-c}}$ for the classes of graph databases $\mathcal{G}$ such that $\mathsf{Hom}(\mathcal{G})$ and $\mathsf{RGPHom}(\mathcal{G})$ are NP-complete. Then:

Proposition III.1. *Unless* P=NP*, it is the case that:*

$$\mathsf{Hom}_{\text{NP-c}} \subsetneq \mathsf{RGPHom}_{\text{NP-c}}.$$

Proof. The separating example $\mathcal{G}$ is a directed cycle on three elements (encoded as the graph database $\mathcal{G} = \{u_1 \overset{a}{\rightsquigarrow} u_2, u_2 \overset{a}{\rightsquigarrow} u_3, u_3 \overset{a}{\rightsquigarrow} u_1\}$). Then $\mathsf{Hom}(\mathcal{G})$ is tractable. In fact, $\mathcal{G}$ has *bounded width*, a property that ensures tractability of $\mathsf{Hom}(\mathcal{G})$ [3]. In turn, $\mathsf{RGPHom}(\mathcal{G})$ is NP-complete. Indeed, consider the regular expression $L = a + aa$. Then $L(\mathcal{G})$ is the *inequality* relation over $\mathsf{Dom}(\mathcal{G})$, i.e., $L(\mathcal{G}) = \{(u_i, u_j) \mid 1 \leq i, j \leq 3 \text{ and } i \neq j\}$. We can reduce then 3-Colorability to $\mathsf{RGPHom}(\mathcal{G})$: Given an undirected graph $G = (V, E)$, construct an RGP $\mathcal{P}_G$ whose set of edges is $\{u \overset{L}{\rightsquigarrow} v \mid (u, v) \in E\}$. It is clear that G is 3-colorable iff $\mathcal{P}_G \to \mathcal{G}$. □

Hence, we cannot directly reduce the study of $\mathsf{RGPHom}(\mathcal{G})$ to $\mathsf{Hom}(\mathcal{G})$. To better understand the complexity of $\mathsf{RGPHom}(\mathcal{G})$ then, we develop a connection with the usual homomorphism problem based on the notion of *regular expansions* of graph databases. Let us reinforce the fact that in the rest of the section we focus on RGPs only, i.e., no inverses are allowed. 2RGPs are studied in Section III-B.

Regular expansions: Let $\mathcal{G}$ be a graph database over Σ. A set $S \subseteq \mathsf{Dom}(\mathcal{G}) \times \mathsf{Dom}(\mathcal{G})$ is $\mathsf{Reg}(\Sigma)$-*definable*, if there is a regular expression L over Σ such that $S = L(\mathcal{G})$.

Definition III.1 (Regular expansions)**.** *Let $\mathcal{G}$ be a graph database over Σ, and suppose that $S_1, \ldots, S_\ell$ is an enumeration of all $\mathsf{Reg}(\Sigma)$-definable subsets of $\mathsf{Dom}(\mathcal{G}) \times \mathsf{Dom}(\mathcal{G})$. The* regular expansion *$\mathcal{E}_\mathcal{G}$ of $\mathcal{G}$ is a graph database over alphabet $\{s_1, \ldots, s_\ell\}$, where each s_i is a fresh symbol, whose set of edges is $\{u \overset{s_i}{\rightsquigarrow} v \mid (u, v) \in S_i\}$.*

The crucial property of the regular expansion $\mathcal{E}_\mathcal{G}$ is that it defines a problem $\mathsf{Hom}(\mathcal{E}_\mathcal{G})$ which is polynomially interreducible with $\mathsf{RGPHom}(\mathcal{G})$.

Theorem III.2. *$\mathsf{RGPHom}(\mathcal{G})$ and $\mathsf{Hom}(\mathcal{E}_\mathcal{G})$ are polynomially interreducible, for each graph database $\mathcal{G}$.*

Proof. Let $S_1, \ldots, S_\ell$ be the enumeration of the $\mathsf{Reg}(\Sigma)$-definable sets used in the construction of $\mathcal{E}_\mathcal{G}$, and $E_1, \ldots, E_\ell$ their corresponding witnessing regular expressions. First we show that $\mathsf{Hom}(\mathcal{E}_\mathcal{G})$ reduces to $\mathsf{RGPHom}(\mathcal{G})$. Let $\mathcal{H}$ be a graph database over $\{s_1, \ldots, s_\ell\}$, and $\mathcal{P}_\mathcal{H}$ be the RGP obtained from $\mathcal{H}$ by replacing each edge $u \overset{s_i}{\rightsquigarrow} v$, for $1 \leq i \leq \ell$, with $u \overset{E_i}{\rightsquigarrow} v$. Clearly, $\mathcal{H} \to \mathcal{E}_\mathcal{G} \Leftrightarrow \mathcal{P}_\mathcal{H} \to \mathcal{G}$. On the other hand, let $\mathcal{P}$ be an RGP and $\mathcal{H}_\mathcal{P}$ be the graph database that is obtained from $\mathcal{P}$ by replacing each edge $u \overset{L}{\rightsquigarrow} v$ by $u \overset{s_{i^*}}{\rightsquigarrow} v$, where $1 \leq i^* \leq \ell$ satisfies that $S_{i^*} = L(\mathcal{G})$. Since each $L(\mathcal{G})$ can be constructed in polynomial time, it follows that $\mathcal{H}_\mathcal{P}$ can also be constructed in polynomial time. Clearly, $\mathcal{P} \to \mathcal{G} \Leftrightarrow \mathcal{H}_\mathcal{P} \to \mathcal{E}_\mathcal{G}$. □

A consequence of the previous theorem is that establishing a dichotomy for $\mathsf{RGPHom}(\mathcal{G})$ reduces to establishing a dichotomy for the usual homomorphism problem over a particular class of templates; namely, the ones that correspond to regular expansions of graph databases:

Corollary III.3. *The following are equivalent:*

1) *For each graph database $\mathcal{G}$ the problem $\mathsf{RGPHom}(\mathcal{G})$ is tractable or* NP*-complete.*
2) *For each regular expansion $\mathcal{E}_\mathcal{G}$ the problem $\mathsf{Hom}(\mathcal{E}_\mathcal{G})$ is tractable or* NP*-complete.*

The Algebraic Dichotomy Conjecture: The currently held dichotomy conjecture for the standard non-uniform homomorphism problem is based on algebraic notions. For such a reason, it is known as the *Algebraic Dichotomy Conjecture* (ADC). Based on the ADC, we develop an algebraic dichotomy conjecture for our problem $\mathsf{RGPHom}(\mathcal{G})$, which holds if the ADC holds. To do this, we need to further refine

the previously established connection between $\mathsf{RGPHom}(\mathcal{G})$ and the usual non-uniform homomorphism problem $\mathsf{Hom}(\mathcal{G})$.

For simplicity, we state the ADC only for graph databases. This is without loss of generality, as the general ADC can be reduced to its version over graph databases [27]. A central notion for the ADC is that of *polymorphism*, as defined below. Let $\mathcal{G}$ be a graph database over Σ. A polymorphism of $\mathcal{G}$ is a mapping $f : \mathsf{Dom}(\mathcal{G})^n \to \mathsf{Dom}(\mathcal{G})$, for some integer $n \geq 1$, such that for each symbol $a \in \Sigma$ and set $\{u_i \overset{a}{\rightsquigarrow} v_i \mid 1 \leq i \leq n\}$ of edges in $\mathcal{G}$, it is the case that $f(u_1, \ldots, u_n) \overset{a}{\rightsquigarrow} f(v_1, \ldots, v_n)$ is also in $\mathcal{G}$.

Crucial to the ADC are the so-called *weak near-unanimity* (WNU) polymorphisms, which are polymorphisms satisfying certain identities (for a definition see e.g. [4], [5]). Also, the ADC is usually stated in terms of *cores* [32]. A graph database $\mathcal{G}$ is a core if every homomorphism $h : \mathcal{G} \to \mathcal{G}$ is surjective. It is known that when classifying the complexity of $\mathsf{Hom}(\mathcal{G})$ we can assume without loss of generality that $\mathcal{G}$ is a core.

Conjecture III.4 (ADC)**.** *Let $\mathcal{G}$ be a graph database that is a core. Then $\mathsf{Hom}(\mathcal{G})$ is tractable if $\mathcal{G}$ has a WNU polymorphism, and it is NP-complete otherwise.*

Let us remark that only the first part of the ADC remains open: it is known that if $\mathcal{G}$ is a core without WNU polymorphisms, then $\mathsf{Hom}(\mathcal{G})$ is NP-complete [4], [5].

Now we present our conjecture for the non-uniform RGP homomorphism problem. To do so, we introduce a notion of polymorphism that plays a similar role for $\mathsf{RGPHom}(\mathcal{G})$ than the usual notion for $\mathsf{Hom}(\mathcal{G})$. Let $\mathcal{G}$ be a graph database over Σ. For nodes u, v of $\mathcal{G}$, we denote by $L_{\mathcal{G}}(u, v)$ the set of words over Σ that can be "read" from u to v in $\mathcal{G}$; i.e.:

$$L_{\mathcal{G}}(u, v) = \{\lambda(\rho) \mid \rho \text{ is a path from } u \text{ to } v \text{ in } \mathcal{G}\}.$$

A *regular polymorphism* of $\mathcal{G}$ is a mapping $f : \mathsf{Dom}(\mathcal{G})^n \to \mathsf{Dom}(\mathcal{G})$, for $n \geq 1$, such that for each tuples $\bar{u} = (u_1, \ldots, u_n)$ and $\bar{v} = (v_1, \ldots, v_n)$ in $\mathsf{Dom}(\mathcal{G})^n$, it is the case that:

$$L_{\mathcal{G}}(u_i, v_i) \subseteq L_{\mathcal{G}}(f(\bar{u}), f(\bar{v})), \quad \text{for some } 1 \leq i \leq n.$$

It is worth noticing that the notion of regular polymorphism has been applied before to study CRPQ-definability [28]. This is in line with the well-studied relationship between the usual notion of polymorphism and CQ-definability [4], [5], [33].

We now postulate our conjecture. Without loss of generality, we only state it in terms of cores of graph databases.

Conjecture III.5 (ADC for RGPs)**.** *Let $\mathcal{G}$ be a graph database that is a core. Then $\mathsf{RGPHom}(\mathcal{G})$ is tractable if $\mathcal{G}$ has a WNU regular polymorphism, and it is NP-complete otherwise.*

The key property of regular polymorphisms that supports our conjecture is established below. This enables us to connect some relevant algebraic properties of $\mathcal{G}$ with those of $\mathcal{E}_{\mathcal{G}}$:

Lemma III.6. *Let $\mathcal{G}$ be a graph database. The following are equivalent for each mapping $f : \mathsf{Dom}(\mathcal{G})^n \to \mathsf{Dom}(\mathcal{G})$:*

- *f is a regular polymorphism of $\mathcal{G}$.*
- *f is a polymorphism of $\mathcal{E}_{\mathcal{G}}$.*

Proof. Recall that $\mathcal{E}_{\mathcal{G}}$ is defined over the alphabet $\{s_1, \ldots, s_\ell\}$, given by an enumeration $S_1, \ldots, S_\ell$ of the $\mathsf{Reg}(\Sigma)$-definable sets of $\mathcal{G}$. Let $E_1, \ldots, E_\ell$ be the corresponding witnessing regular expressions. Assume first that $f : \mathsf{Dom}(G)^n \to \mathsf{Dom}(\mathcal{G})$ is not a polymorphism of $\mathcal{E}_{\mathcal{G}}$. Then there is a symbol s_i, with $1 \leq i \leq \ell$, and tuples $\bar{u} = (u_1, \ldots, u_n)$, $\bar{v} = (v_1, \ldots, v_n)$ such that (a) $u_j \overset{s_i}{\rightsquigarrow} v_j \in \mathcal{E}_{\mathcal{G}}$, for each $1 \leq j \leq n$, and (b) $f(\bar{u}) \overset{s_i}{\rightsquigarrow} f(\bar{v}) \notin \mathcal{E}_{\mathcal{G}}$. Therefore, $(u_j, v_j) \in S_i$, for each $1 \leq j \leq n$, and thus there is a word w_j in $L_{\mathcal{G}}(u_j, v_j)$ satisfying E_i. Notice that $w_j \notin L_{\mathcal{G}}(f(\bar{u}), f(\bar{v}))$, for each $1 \leq j \leq n$; otherwise $(f(\bar{u}), f(\bar{v})) \in S_i$, and hence $f(\bar{u}) \overset{s_i}{\rightsquigarrow} f(\bar{v})$ would belong to $\mathcal{E}_{\mathcal{G}}$. We conclude that f is not a regular polymorphism of $\mathcal{G}$. Assume now that f is not a regular polymorphism of $\mathcal{G}$. Then there are tuples $\bar{u} = (u_1, \ldots, u_n)$, $\bar{v} = (v_1, \ldots, v_n)$, and words $w_1, \ldots, w_n$ such that: (i) $w_j \in L_{\mathcal{G}}(u_j, v_j)$ for each $1 \leq j \leq n$, and (ii) $w_j \notin L_{\mathcal{G}}(f(\bar{u}), f(\bar{v}))$, for each $1 \leq j \leq n$. Consider the regular expression $E = w_1 + \cdots + w_n$, and pick an $i^* \in \{1, \ldots, \ell\}$ such that $S_{i^*} = E(\mathcal{G})$. Then $(u_j, v_j) \in S_{i^*}$, and thus $u_j \overset{s_{i^*}}{\rightsquigarrow} v_j \in \mathcal{E}_{\mathcal{G}}$, for each $1 \leq j \leq n$. By condition (ii), we have that $(f(\bar{u}), f(\bar{v})) \notin S_{i^*}$, and then $f(\bar{u}) \overset{s_{i^*}}{\rightsquigarrow} f(\bar{v}) \notin \mathcal{E}_{\mathcal{G}}$. Hence, f is not a polymorphism of $\mathcal{E}_{\mathcal{G}}$. □

As a corollary to Lemma III.6 we obtain that the ADC implies the ADC for RGPs:

Corollary III.7. *If ADC holds, then ADC for RGPs holds.*

We do not know if the converse is true. This is equivalent to the question of whether the ADC can be reduced to its version over regular expansions, i.e., graph databases of the form $\mathcal{E}_{\mathcal{G}}$.

Let us finish by observing that only the first part of the ADC for RGPs remains open (as for the ADC); i.e., whether the existence of a WNU regular polymorphism of $\mathcal{G}$ implies tractability of $\mathsf{RGPHom}(\mathcal{G})$. In fact:

Proposition III.8. *Let $\mathcal{G}$ be a core without WNU regular polymorphisms. Then $\mathsf{RGPHom}(\mathcal{G})$ is NP-complete.*

Proof. It is straightforward to prove that $\mathcal{E}_{\mathcal{G}}$ is also a core. In view of Lemma III.6, $\mathcal{E}_{\mathcal{G}}$ does not have a WNU polymorphism, and hence $\mathsf{Hom}(\mathcal{E}_{\mathcal{G}})$ is NP-complete. From Theorem III.2, we conclude then that $\mathsf{RGPHom}(\mathcal{G})$ is NP-complete. □

B. Adding inverses

An analogous ADC can also be stated for the problem $\mathsf{2RGPHom}(\mathcal{G})$ – i.e., when we add inverses to RGPs. The only difference is that now tractability is stated in terms of *two-way* regular polymorphisms. These are defined exactly as before, save that now we work over the completion $\mathcal{G}^{\pm}$ of $\mathcal{G}$.

However, it might be simpler to actually establish a dichotomy in this case, as $\mathsf{2RGPHom}(\mathcal{G})$ easily becomes NP-complete. This suggests that the tractability cases must be quite restricted. Our next result confirms this claim and establishes a complete dichotomy for the case of directed graphs.

Theorem III.9. *Let $\mathcal{G}$ be a directed graph that is a core. Then $\mathsf{2RGPHom}(\mathcal{G})$ is tractable if $\mathcal{G}$ is either (i) the directed path of*

length 1*, (ii) the directed cycle of length* 1*, or (iii) the directed cycle of length* 2*. Otherwise,* $\mathsf{2RGPHom}(\mathcal{G})$ *is NP-complete.*

It is worth noticing that this is true only because inverses are available. In particular, Theorem III.9 does not hold for $\mathsf{RGPHom}(\mathcal{G})$. Moreover, there are some simple directed graphs $\mathcal{G}$ for which $\mathsf{RGPHom}(\mathcal{G})$ is tractable, but $\mathsf{2RGPHom}(\mathcal{G})$ is NP-complete. These include, for instance, all directed paths of length at least two.

Connections with non-uniform evaluation for $\exists^+\mathrm{FO}$*:* We now establish a connection between $\mathsf{2RGPHom}(\mathcal{G})$ and $\exists^+\mathrm{FO}(\mathcal{G})$, i.e., the problem of evaluating Boolean existential positive FO formulas over fixed template $\mathcal{G}$. Let us notice first that for each graph database $\mathcal{G}$ the problem $\mathsf{2RGPHom}(\mathcal{G})$ can be reduced in polynomial time to the problem $\exists^+\mathrm{FO}(\mathcal{G})$:

Proposition III.10. *Let* $\mathcal{G}$ *be a fixed graph database. There is a polynomial time reduction that on input a 2RGP* $\mathcal{P}$ *constructs an* $\exists^+\mathrm{FO}$ *sentence* $\theta_{\mathcal{P}}$ *such that* $\mathcal{P} \to \mathcal{G} \Leftrightarrow \mathcal{G} \models \theta_{\mathcal{P}}$.

It is worth understanding whether a polynomial time reduction in the opposite direction can also be obtained. This is important since a dichotomy for the non-uniform problem $\exists^+\mathrm{FO}(\mathcal{G})$ is known to hold [13], [14], and thus such a reduction would provide us with a dichotomy for $\mathsf{2RGPHom}(\mathcal{G})$ as well. We prove next that it is not possible to find such a reduction (under complexity theoretical assumptions). Following previous terminology, we write $\exists^+\mathrm{FO}_{\text{NP-c}}$ for the class of graph databases $\mathcal{G}$ such that $\exists^+\mathrm{FO}(\mathcal{G})$ is NP-complete. Then:

Proposition III.11. *Unless* P=NP*, it is the case that* $\mathsf{2RGPHom}_{\text{NP-c}} \subsetneq \exists^+\mathrm{FO}_{\text{NP-c}}$.

The separating example consists of a single directed edge between two different nodes.

IV. Structural Restrictions for the 2RGP Homomorphism Problem

Let $\mathbb{C}$ be a class of 2RGPs. Recall that $\mathsf{2RGPHom}(\mathbb{C}, -)$ is the problem of checking if $\mathcal{P} \to \mathcal{G}$, given a 2RGP $\mathcal{P} \in \mathbb{C}$ and a graph database $\mathcal{G}$. We study which classes $\mathbb{C}$ of 2RGPs ensure that $\mathsf{2RGPHom}(\mathbb{C}, -)$ can be efficiently solved. In particular, we concentrate on studying how the notion of bounded treewidth modulo equivalence relates to the efficient solvability of $\mathsf{2RGPHom}(\mathbb{C}, -)$. Based on the fact that there is no *canonical* way to define "homomorphic equivalence for 2RGPs", we propose two natural interpretations of the notion – one based on a suitable class of homomorphisms for 2RGPs and the other one on the logical equivalence of their underlying C2RPQs – and prove that they both yield classes of 2RGPs of bounded treewidth modulo equivalence with good properties in terms of the 2RGP homomorphism problem.

A. Bounded treewidth

The homomorphism problem $\mathsf{Hom}(\mathbb{C}, -)$ can be solved in polynomial time when $\mathbb{C}$ is a class of graph databases of bounded treewidth. This can be proved by using techniques based on the *existential pebble game* [11]. Such techniques can naturally be extended to also show that $\mathsf{2RGPHom}(\mathbb{C}, -)$ is tractable when $\mathbb{C}$ is a class of 2RGPs of bounded treewidth. We present such techniques in this section.

Recall that the *treewidth* of a graph is a measure of how much the graph resembles a tree; cf. [34]. This is formalized using the notion of *tree decomposition*, as follows. Let $G = (V, E)$ be an undirected graph. A tree decomposition of G is a pair (T, β), where T is a tree and β is a mapping that assigns a nonempty set of nodes in V to each node t in T, such that:

1) For each $v \in V$ it is the case that the set of nodes $t \in T$ such that $v \in \beta(t)$ is connected.
2) For each edge $\{u, v\} \in E$ there is a node $t \in T$ such that $\{u, v\} \subseteq \beta(t)$.

The *width* of (T, β) is $\max\{|\beta(t)| \mid t \in T\} - 1$. The treewidth of G is the minimum width of its tree decompositions.

The treewidth of a graph database $\mathcal{G} = \{u_i \overset{a_i}{\rightsquigarrow} v_i \mid 1 \leq i \leq m\}$ over Σ is the treewidth of its underlying undirected graph $(\mathsf{Dom}(\mathcal{G}), \{(u_i, v_i) \mid 1 \leq i \leq m\})$ The treewidth of a 2RGP $\mathcal{P}$ over Σ is thus the treewidth of the graph database $\mathcal{G}$ over $\mathsf{Reg}(\Sigma^{\pm})$ that represents it. For $k \geq 1$, we write $\mathsf{TW}(k)$ for the class of graph databases of treewidth at most k, and $\mathsf{TW}(k)_{\text{2rgp}}$ for the class of 2RGPs of treewidth at most k.

The existential pebble game for 2RGPs: We present a natural extension of the *existential pebble game* [26] to handle the semantics of 2RGPs. Fix $k \geq 1$. Let $\mathcal{P}$ and $\mathcal{G}$ be a 2RGP and a graph database over Σ, respectively. The existential k-pebble game on $(\mathcal{P}, \mathcal{G})$ proceeds in rounds. In the first round Spoiler places his pebbles $\mathtt{p}_1, \ldots, \mathtt{p}_k$ on (not necessarily distinct) elements $c_1, \ldots, c_k$ of $\mathsf{Dom}(\mathcal{P})$, and Duplicator responds by placing her pebbles $\mathtt{q}_1, \ldots, \mathtt{q}_k$ on elements $d_1, \ldots, d_k$ of $\mathsf{Dom}(\mathcal{G})$. In each further round, Spoiler removes one of his pebbles, say $\mathtt{p}_i$, for $1 \leq i \leq k$, and places it on an element of $\mathsf{Dom}(\mathcal{P})$, and Duplicator responds by placing her corresponding pebble $\mathtt{q}_i$ on an element of $\mathsf{Dom}(\mathcal{G})$. Duplicator wins if she has a *winning strategy*, i.e., she can indefinitely continue playing the game in such way that after each round, if $c_1, \ldots, c_k$ and $d_1, \ldots, d_k$ are the elements covered by pebbles $\mathtt{p}_1, \ldots, \mathtt{p}_k$ and $\mathtt{q}_1, \ldots, \mathtt{q}_k$ in $\mathcal{P}$ and $\mathcal{G}$, respectively, then: $\big((c_1, \ldots, c_k), (d_1, \ldots, d_k)\big)$ satisfies that for every edge $c \overset{L}{\rightsquigarrow} c' \in \mathcal{P}$, where L is a regular expression over $\Sigma^{\pm}$ and c, c' appear in $(c_1, \ldots, c_k)$, we have that $(d, d') \in L(\mathcal{G})$, where d and d' are the elements corresponding to c and c', respectively, in $(d_1, \ldots, d_k)$. We write $\mathcal{P} \to_k \mathcal{G}$ if Duplicator wins.

The following two results establish the good properties of the existential k-pebble game in terms of the 2RGP homomorphism problem for the class $\mathsf{TW}(k)$:

Proposition IV.1 (Follows from [11])**.** *For each* $k \geq 1$*, if* $\mathcal{P}$ *is a 2RGP in* $\mathsf{TW}(k)_{\text{2rgp}}$ *and* $\mathcal{G}$ *is a graph database, then:*

$$\mathcal{P} \to_{k+1} \mathcal{G} \quad \Longleftrightarrow \quad \mathcal{P} \to \mathcal{G}.$$

In addition:

Proposition IV.2 (Follows from [26])**.** *Fix* $k \geq 1$*. The problem of checking if* $\mathcal{P} \to_k \mathcal{G}$*, for a 2RGP* $\mathcal{P}$ *and a graph database* $\mathcal{G}$*, can be solved in time* $(|\mathcal{P}| + |\mathcal{G}|)^{O(k)}$.

Thus, to check whether $\mathcal{P} \to \mathcal{G}$ for $\mathcal{P} \in \mathsf{TW}(k)_{\mathrm{2rgp}}$, it is sufficient to check in time $(|\mathcal{P}|+|\mathcal{G}|)^{O(k)}$ if $\mathcal{P} \to_{k+1} \mathcal{G}$. This yields the tractability of $\mathsf{2RGPHom}(\mathsf{TW}(k)_{\mathrm{2rgp}}, -)$:

Theorem IV.3. *Fix $k \geq 1$. Then $\mathsf{2RGPHom}(\mathsf{TW}(k)_{\mathrm{2rgp}}, -)$ can be solved in polynomial time $(|\mathcal{P}|+|\mathcal{G}|)^{O(k)}$.*

B. Bounded treewidth modulo homomorphic equivalence

We start by recalling the notion of bounded treewidth modulo (homomorphic) equivalence for graph databases. We write $\mathsf{TW}(k)_{\leftrightarrows}$ for the class of graph databases $\mathcal{G}$ that are homomorphically equivalent to some $\mathcal{G}'$ in $\mathsf{TW}(k)$; i.e., $\mathcal{G} \to \mathcal{G}'$ and $\mathcal{G}' \to \mathcal{G}$ (written $\mathcal{G} \leftrightarrows \mathcal{G}'$). It is easy to see that this notion properly extends bounded treewidth; i.e., $\mathsf{TW}(k) \subsetneq \mathsf{TW}(k)_{\leftrightarrows}$ for each $k \geq 1$, and $\mathsf{TW}(k)_{\leftrightarrows} \not\subseteq \mathsf{TW}(\ell)$ for each $\ell \geq 1$.

The reason why the homomorphism problem for the classes of graph databases of bounded treewidth modulo equivalence can be solved in polynomial time is simple: the existential pebble game techniques developed above continue to apply for them. Formally, let $\mathcal{G}$ and $\mathcal{G}'$ be graph databases such that $\mathcal{G}$ is in $\mathsf{TW}(k)_{\leftrightarrows}$. Then checking if $\mathcal{G} \to \mathcal{G}'$ still boils down to checking if $\mathcal{G} \to_{k+1} \mathcal{G}'$ [11]. Hence, $\mathsf{Hom}(\mathsf{TW}(k)_{\leftrightarrows}, -)$ can be solved in time $(|\mathcal{G}|+|\mathcal{G}'|)^{O(k)}$.

The goal of this section is to present a simple, yet meaningful notion of "homomorphic equivalence for 2RGPs" that preserves such a good behavior for the problem $\mathsf{2RGPHom}(\mathbb{C}, -)$. The notion is based on a class of homomorphisms between 2RGPs that we define next.

2RGP homomorphisms: Our definition uses the notion of containment between regular languages. However, since our regular expressions can mention inverses, it is convenient to work with a more flexible notion of containment based on *foldings* [24]. Let Σ be a finite alphabet. Recall that we denote by $\Sigma^{\pm}$ the alphabet $\Sigma \cup \{a^- \mid a \in \Sigma\}$. If $p \in \Sigma^{\pm}$ and $p = a$ for some $a \in \Sigma$, then p^- denotes a^-. On the other hand, if $p = a^-$ for $a \in \Sigma$, then p^- denotes a. Let $s = s_1 \dots s_k$ and $t = t_1 \dots t_\ell$ be words over $\Sigma^{\pm}$. Then t *folds* onto s if there is a sequence $i_0, \dots, i_\ell$ of positions in $\{0, \dots, k\}$ such that:

- $i_0 = 0$ and $i_\ell = k$, and
- for each $1 \leq j \leq \ell$, it is the case that $i_j = i_{j-1} + 1$ and $t_j = s_{i_j}$, or $i_j = i_{j-1} - 1$ and $t_j = s_{i_{j-1}}^-$.

Intuitively, t folds onto s if t can be read in s by a *two-way automaton* that outputs symbol p, each time p is read from left-to-right, and p^-, each time p is read from right-to-left. For instance, $abb^-a^-abb^-c$ folds into abb^-c.

If L is a language over $\Sigma^{\pm}$, we write $\mathsf{fold}(L)$ for the set of words u over $\Sigma^{\pm}$ such that some $v \in L$ folds onto u. We can now define our notion of homomorphism between 2RGPs:

Definition IV.1 (2RGP homomoprhisms)**.** *Let $\mathcal{P}$ and $\mathcal{P}'$ be 2RGPs. A* 2RGP homomorphism *from $\mathcal{P}$ to $\mathcal{P}'$ is a mapping $r : \mathsf{Dom}(\mathcal{P}) \to \mathsf{Dom}(\mathcal{P}')$ such that, for each edge $u \overset{L}{\rightsquigarrow} v$ in $\mathcal{P}$ there is an edge $r(u) \overset{L'}{\rightsquigarrow} r(v)$ in $\mathcal{P}'$, where $L, L' \in \mathsf{Reg}(\Sigma^{\pm})$, such that $L' \subseteq \mathsf{fold}(L)$. We write $\mathcal{P} \to_{\mathrm{2rgp}} \mathcal{P}'$ if there is a 2RGP homomorphism from $\mathcal{P}$ to $\mathcal{P}'$, and $\mathcal{P} \leftrightarrows_{\mathrm{2rgp}} \mathcal{P}'$ if $\mathcal{P} \to_{\mathrm{2rgp}} \mathcal{P}'$ and $\mathcal{P}' \to_{\mathrm{2rgp}} \mathcal{P}$.* □

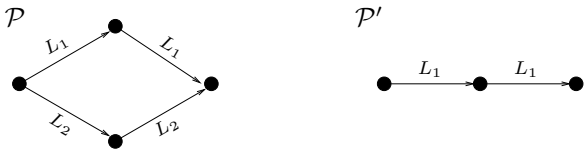

Fig. 1. The 2RGPs $\mathcal{P}$ and $\mathcal{P}'$ from Example IV.4.

Let $k \geq 1$. We write $\mathsf{TW}(k)_{\leftrightarrows_{\mathrm{2rgp}}}$ for the class of 2RGPs $\mathcal{P}$ such that $\mathcal{P} \leftrightarrows_{\mathrm{2rgp}} \mathcal{P}'$ for some $\mathcal{P}'$ in $\mathsf{TW}(k)_{\mathrm{2rgp}}$.

Example IV.4. Consider regular languages L_1, L_2 over Σ such that $L_1 \subseteq L_2$. Let $\mathcal{P}$ be the 2RGP shown in Figure 1. Clearly, $\mathcal{P} \in \mathsf{TW}(2)_{\mathrm{2rgp}} \setminus \mathsf{TW}(1)_{\mathrm{2rgp}}$. On the other hand, $\mathcal{P}$ is in $\mathsf{TW}(1)_{\leftrightarrows_{\mathrm{rgp}}}$. In fact, it is easy to see that $\mathcal{P} \leftrightarrows_{\mathrm{2rgp}} \mathcal{P}'$, where $\mathcal{P}'$ is the 2RGP in $\mathsf{TW}(1)_{\mathrm{rgp}}$ also shown in Figure 1. □

In accordance with the previous example, we notice that bounded treewidth modulo 2RGP homomorphic equivalence properly extends bounded treewidth for 2RGPs:

Proposition IV.5. *Let $k \geq 1$. Then $\mathsf{TW}(k)_{\mathrm{2rgp}} \subsetneq \mathsf{TW}(k)_{\leftrightarrows_{\mathrm{2rgp}}}$ and $\mathsf{TW}(k)_{\leftrightarrows_{\mathrm{2rgp}}} \not\subseteq \mathsf{TW}(\ell)_{\mathrm{2rgp}}$ for each $\ell \geq 1$.*

We prove next that $\mathsf{2RGPHom}(\mathsf{TW}(k)_{\leftrightarrows_{\mathrm{2rgp}}}, -)$ can be solved in polynomial time by using the existential $(k+1)$-pebble game for 2RGPs:

Theorem IV.6. *Fix $k \geq 1$. Then for each 2RGP $\mathcal{P}$ in $\mathsf{TW}(k)_{\leftrightarrows_{\mathrm{2rgp}}}$ and graph database $\mathcal{G}$ it is the case that:*

$$\mathcal{P} \to \mathcal{G} \quad \Longleftrightarrow \quad \mathcal{P} \to_{k+1} \mathcal{G}.$$

Therefore, $\mathsf{2RGPHom}(\mathsf{TW}(k)_{\leftrightarrows_{\mathrm{2rgp}}}, -)$ can be solved in polynomial time $(|\mathcal{P}|+|\mathcal{G}|)^{O(k)}$.

Proof. Clearly, if $\mathcal{P} \to \mathcal{G}$ then $\mathcal{P} \to_{k+1} \mathcal{G}$ (the Duplicator simply responds by following the homomorphism). Assume, on the other hand, that $\mathcal{P} \to_{k+1} \mathcal{G}$. Since $\mathcal{P}$ is in $\mathsf{TW}(k)_{\leftrightarrows_{\mathrm{2rgp}}}$, there is a 2RGP $\mathcal{P}'$ in $\mathsf{TW}(k)_{\mathrm{2rgp}}$ such that $\mathcal{P} \leftrightarrows_{\mathrm{2rgp}} \mathcal{P}'$. Thus, $\mathcal{P}' \to_{\mathrm{2rgp}} \mathcal{P}$. It is not hard to prove that, together with the fact that $\mathcal{P} \to_{k+1} \mathcal{G}$, this implies $\mathcal{P}' \to_{k+1} \mathcal{G}$, and hence $\mathcal{P}' \to \mathcal{G}$ from Proposition IV.1 since $\mathcal{P}'$ is in $\mathsf{TW}(k)_{\mathrm{2rgp}}$. But we also have that $\mathcal{P} \to_{\mathrm{2rgp}} \mathcal{P}'$. It is not hard to prove that, together with the fact that $\mathcal{P}' \to \mathcal{G}$, this implies that $\mathcal{P} \to \mathcal{G}$. □

Decidability of the notion: The problem of checking if a 2RGP $\mathcal{P}$ is in $\mathsf{TW}(k)_{\leftrightarrows_{\mathrm{2rgp}}}$ is decidable. More precisely:

Theorem IV.7. *For each fixed $k \geq 1$, it is* PSPACE*-complete to check whether a 2RGP $\mathcal{P}$ is in $\mathsf{TW}(k)_{\leftrightarrows_{\mathrm{2rgp}}}$.*

This is in contrast with the problem of checking bounded treewidth modulo homomorphic equivalence for graph databases, which is NP-complete. More formally, for each fixed $k \geq 1$ it is NP-complete to check if the graph database $\mathcal{G}$ is in $\mathsf{TW}(k)_{\leftrightarrows}$ [11]. The difference lies in the fact that checking $\mathcal{G} \leftrightarrows \mathcal{G}'$, for graph databases $\mathcal{G}$ and $\mathcal{G}'$, is in NP, but this no longer holds if we want to check $\mathcal{P} \leftrightarrows_{\mathrm{2rgp}} \mathcal{P}'$ for 2RGPs $\mathcal{P}$ and $\mathcal{P}'$. In fact, this step requires checking

containment between regular expressions, which is a PSPACE-complete problem [35].

Implication for C2RPQ evaluation: As mentioned in Section II, each Boolean C2RPQ ϕ can be associated with a 2RGP $\mathcal{P}_\phi$ such that $\mathcal{G} \models \phi \Leftrightarrow \mathcal{P}_\phi \to \mathcal{G}$ for each graph database $\mathcal{G}$. This observation, together with our previous results, allows us to identify a large class of C2RPQs for which evaluation is tractable. Formally, let $\mathsf{TW}(k)^{\phi}_{\leftrightarrows_{2\mathrm{rgp}}}$ be the class of C2RPQs ϕ such that its associated 2RGP $\mathcal{P}_\phi$ is in $\mathsf{TW}(k)_{\leftrightarrows_{2\mathrm{rgp}}}$. Then:

Corollary IV.8. *Fix $k \geq 1$. Evaluation for C2RPQs in $\mathsf{TW}(k)^{\phi}_{\leftrightarrows_{2\mathrm{rgp}}}$ can be solved in polynomial time $(|\phi|+|\mathcal{G}|)^{O(k)}$.*

C. Bounded treewidth modulo logical equivalence

In this section we stress the connection between 2RGPs and C2RPQs even further. It is worth, however, to start by recalling an important connection between the notion of homomorphic equivalence for graph databases and the *logical equivalence* of the CQs associated with them.

CQ equivalence: Recall from Section II that each Boolean CQ q over graph databases can be associated with a graph database $\mathcal{G}_q$ – and, correspondingly, each graph database $\mathcal{G}$ can be associated with a Boolean CQ $q_\mathcal{G}$ over graph databases – in such a way that for each graph database $\mathcal{H}$:

$$\mathcal{H} \models q \iff \mathcal{G}_q \to \mathcal{H} \quad \text{and} \quad \mathcal{G} \to \mathcal{H} \iff \mathcal{H} \models q_\mathcal{G}.$$

It is known that the homomorphic equivalence of graph databases $\mathcal{G}$ and $\mathcal{G}'$ can be recast in terms of the equivalence of their associated Boolean CQs $q_\mathcal{G}$ and $q_{\mathcal{G}'}$, respectively. Analogously, the equivalence of Boolean CQs q and q' boils down to the homomorphic equivalence of the graph databases $\mathcal{G}_q$ and $\mathcal{G}_{q'}$, respectively [36]. Formally, Boolean CQs q and q' over graph databases are equivalent, denoted $q \equiv q'$, if for every graph database $\mathcal{G}$ we have that $\mathcal{G} \models q \Leftrightarrow \mathcal{G} \models q'$. Then:

$$\mathcal{G} \leftrightarrows \mathcal{G}' \iff q_\mathcal{G} \equiv q_{\mathcal{G}'} \quad \text{and} \quad q \equiv q' \iff \mathcal{G}_q \leftrightarrows \mathcal{G}_{q'}.$$

Hence, the notion of bounded treewidth modulo equivalence for graph databases can be equivalently expressed in terms of the equivalence of Boolean CQs. Formally, let us write $\mathsf{TW}(k)^{q}_{\leftrightarrows}$ for the class of Boolean CQs q that are equivalent to some q' that is associated with a graph database $\mathcal{G}_{q'} \in \mathsf{TW}(k)$. It is clear then that for each Boolean CQ q with associated graph database $\mathcal{G}_q$, and for each graph database $\mathcal{G}$ with associated Boolean CQ $q_\mathcal{G}$, we have that:

$$q \in \mathsf{TW}(k)^{q}_{\leftrightarrows} \iff \mathcal{G}_q \in \mathsf{TW}(k)_{\leftrightarrows} \quad \text{and} \quad \mathcal{G} \in \mathsf{TW}(k)_{\leftrightarrows} \iff q_\mathcal{G} \in \mathsf{TW}(k)^{q}_{\leftrightarrows}.$$

In particular, it follows from Theorem IV.6 that the evaluation problem for CQs in $\mathsf{TW}(k)^{q}_{\leftrightarrows}$ is tractable.

C2RPQ equivalence: Let us now return to our case of interest: the 2RGP homomorphism problem. As mentioned in Section II, in the same way that Boolean CQs can be associated with graph databases, each Boolean C2RPQ ϕ can be associated with a 2RGP $\mathcal{P}_\phi$ – and, correspondingly, each 2RGP $\mathcal{P}$ can be associated with a Boolean C2RPQ $\phi_\mathcal{P}$ over graph databases – in such a way that for each graph database $\mathcal{H}$:

$$\mathcal{H} \models \phi \iff \mathcal{P}_\phi \to \mathcal{H} \quad \text{and} \quad \mathcal{P} \to \mathcal{H} \iff \mathcal{H} \models \phi_\mathcal{P}.$$

It is thus natural – following the previous observations – to define a notion of bounded treewidth modulo equivalence for 2RGPs based on the logical equivalence of their associated C2RPQs. In fact, such notion has proved to be relevant in the context of evaluation and optimization of C2RPQs over graph databases [23]. We formalize this idea next. As before, we write $\phi \equiv \phi'$, for C2RPQs ϕ and ϕ', if for every graph database $\mathcal{G}$ it is the case that $\mathcal{G} \models \phi \Leftrightarrow \mathcal{G} \models \phi'$. Then:

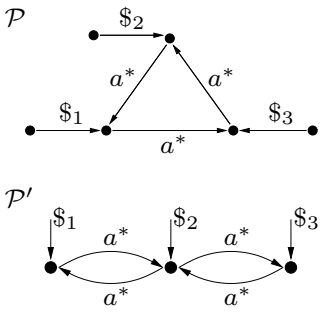

Fig. 2. The 2RGPs $\mathcal{P}$ and $\mathcal{P}'$ from Example IV.9.

Definition IV.2 ($\mathsf{TW}(k)^{\phi}_{\equiv_{2\mathrm{rgp}}}$ and $\mathsf{TW}(k)_{\equiv_{2\mathrm{rgp}}}$)**.** *We define $\mathsf{TW}(k)^{\phi}_{\equiv_{2\mathrm{rgp}}}$, for $k \geq 1$, as the class of Boolean C2RPQs ϕ such that there is a C2RPQ ϕ' with $\phi \equiv \phi'$ and $\mathcal{P}_{\phi'} \in \mathsf{TW}(k)_{2\mathrm{rgp}}$. Analogously, we define $\mathsf{TW}(k)_{\equiv_{2\mathrm{rgp}}}$ as the class of 2RGPs whose associated C2RPQ $\phi_\mathcal{P}$ is in $\mathsf{TW}(k)^{\phi}_{\equiv_{2\mathrm{rgp}}}$.*

Next we provide an example of a 2RGP in $\mathsf{TW}(1)_{\equiv_{2\mathrm{rgp}}}$:

Example IV.9 (Taken from [23])**.** Consider the 2RGP $\mathcal{P}$ shown in Figure 2. It can be proved that $\phi_\mathcal{P} \equiv \phi_{\mathcal{P}'}$, where $\mathcal{P}'$ is also shown in Figure 2. Clearly, $\mathcal{P}' \in \mathsf{TW}(1)_{2\mathrm{rgp}}$, and hence $\mathcal{P} \in \mathsf{TW}(1)_{\equiv_{2\mathrm{rgp}}}$. It can be proved, on the other hand, that $\mathcal{P} \notin \mathsf{TW}(1)_{\leftrightarrows_{2\mathrm{rgp}}}$. □

In line with the previous example, bounded treewidth modulo C2RPQ equivalence properly extends bounded treewidth modulo 2RGP homomorphism equivalence:

Proposition IV.10. *Let $k \geq 1$. Then $\mathsf{TW}(k)_{\leftrightarrows_{2\mathrm{rgp}}} \subsetneq \mathsf{TW}(k)_{\equiv_{2\mathrm{rgp}}}$ and $\mathsf{TW}(k)_{\equiv_{2\mathrm{rgp}}} \not\subseteq \mathsf{TW}(\ell)_{\leftrightarrows_{2\mathrm{rgp}}}$ for each $\ell \geq 1$.*

This extension, on the other hand, comes at a price: It is easy to prove that the characterization of $\mathcal{P} \to \mathcal{G}$ in terms of $\mathcal{P} \to_{k+1} \mathcal{G}$, which holds for the $\mathcal{P}$'s in $\mathsf{TW}(k)_{\leftrightarrows_{2\mathrm{rgp}}}$, no longer holds for the $\mathcal{P}$'s in $\mathsf{TW}(k)_{\equiv_{2\mathrm{rgp}}}$. In addition, it does not even seem possible to modify the existential pebble game techniques to establish good properties of the classes $\mathsf{TW}(k)_{\equiv_{2\mathrm{rgp}}}$ in terms of the 2RGP homomorphism problem.

By using more elaborate techniques based on automata, on the other hand, we are able to show that the classes $\mathsf{TW}(k)_{\equiv_{2\mathrm{rgp}}}$ are well-behaved in terms of $\mathsf{2RGPHom}(\mathbb{C}, -)$. In particular, we prove that the prob-

lem $\mathsf{2RGPHom}(\mathsf{TW}(k)_{\equiv_{2\mathrm{rgp}}}, -)$, for each fixed $k \geq 1$, is *fixed-parameter tractable*, with the parameter being the size of the 2RGP $\mathcal{P}$. Intuitively, this means that $\mathsf{2RGPHom}(\mathsf{TW}(k)_{\equiv_{2\mathrm{rgp}}}, -)$ can be solved by an algorithm whose running time depends only polynomially on the size of $\mathcal{G}$ and more loosely on the size of $\mathcal{P}$. This is a desirable property if we consider that $\mathcal{P}$ is encoding a C2RPQ ϕ, which is in general orders of magnitude smaller than the graph database $\mathcal{G}$. Moreover, under usual complexity theoretical assumptions this good behavior does not extend to all classes of 2RGPs [29]. Our main result establishes the following:

Theorem IV.11. *For each fixed $k \geq 1$ the problem $\mathsf{2RGPHom}(\mathsf{TW}(k)_{\equiv_{2\mathrm{rgp}}}, -)$ can be solved in time:*

$$O\big(|\mathcal{G}|^{t(k)} \cdot 2^{s(|\mathcal{P}|)}\big),$$

for $t : \mathbb{N} \to \mathbb{N}$ a linear function and $s : \mathbb{N} \to \mathbb{N}$ a polynomial.

As a corollary, we thus obtain the following:

Corollary IV.12. *Fix $k \geq 1$. The evaluation problem for C2RPQs in $\mathsf{TW}(k)^{\phi}_{\equiv_{2\mathrm{rgp}}}$ is feasible in time $O(|\mathcal{G}|^{t(k)} \cdot 2^{s(|\phi|)})$, for $t : \mathbb{N} \to \mathbb{N}$ a linear function and $s : \mathbb{N} \to \mathbb{N}$ a polynomial.*

Proof of Theorem IV.11: The proof is based on a small-witness property for the fact that a 2RGP $\mathcal{P}$ is of bounded treewidth modulo C2RPQ equivalence. As a matter of fact, in order to prove Theorem IV.7, we already obtained one such a small-witness property – in fact, a polynomial-witness property – for the classes $\mathsf{TW}(k)_{\leftrightarrows_{2\mathrm{rgp}}}$ of bounded treewidth modulo 2RGP homomorphic equivalence. The small-witness property we establish here is, on the other hand, much more difficult to prove. This is because the fact that a 2RGP $\mathcal{P}$ is in $\mathsf{TW}(k)_{\equiv_{2\mathrm{rgp}}}$ can be caused by some intricate interactions among the regular expressions of $\mathcal{P}$. To handle such complications we need to be more flexible in terms of which objects are allowed to be a "witness" for the fact that 2RGP $\mathcal{P}$ is of bounded treewidth modulo C2RPQ equivalence. In fact, as explained next we allow such witnesses to be expressed as sets of 2RGPs and also to slightly increase the original treewidth considered.

Given a set $\{\mathcal{P}_1, \ldots, \mathcal{P}_n\}$ of 2RGPs and a graph database $\mathcal{G}$, we write $\bigcup_{1 \leq i \leq n} \mathcal{P}_i \to \mathcal{G}$ if $\mathcal{P}_i \to \mathcal{G}$ for some $1 \leq i \leq n$. The next lemma establishes our desired small-witness property.

Lemma IV.13. *Fix $k \geq 1$. There is a single-exponential time algorithm that on input a 2RGP $\mathcal{P}$ in $\mathsf{TW}(k)_{\equiv_{2\mathrm{rgp}}}$, computes a set $\mathfrak{R}$ of 2RGPs such that:*

1) *Each 2RGP $\mathcal{R} \in \mathfrak{R}$ is in $\mathsf{TW}(2k+1)_{2\mathrm{rgp}}$, and*
2) *for each graph database $\mathcal{G}$: $\mathcal{P} \to \mathcal{G}$ iff $\bigcup_{\mathcal{R} \in \mathfrak{R}} \mathcal{R} \to \mathcal{G}$.*

Proof Sketch: Let us first note that the case $k = 1$ follows directly from [23], thus we focus on the case $k > 1$. Let $\mathsf{A}(\cdot)$ and $\mathsf{E}(\cdot)$ be two fixed functions that map a regular expression to an equivalent *nondeterministic finite automaton* (NFA) and vice versa, respectively. It is well-known that we can take A and E to have an output of at most of polynomial and exponential size, respectively. Let s, s' be states in an NFA M. We denote by $M(s, s')$ the NFA obtained from M by setting the initial state as s and the set of final states as $\{s'\}$.

Let $r \geq 1$. Next we define the notion of *r-subdivision*. The intuition is that an r-subdivision of a 2RGP $\mathcal{Q}$ is a 2RGP obtained by "dividing" each regular expression L in $\mathcal{Q}$ into a sequence of regular expressions $L_1, \ldots, L_\ell$, whose concatenation defines a language that is contained in the language of L. The parameter r bounds the possible length of the sequence $L_1, \ldots, L_\ell$. Formally, an *r-subdivision* of a 2RGP $\mathcal{Q}$ is a 2RGP obtained as follows. Replace each edge $u \overset{L}{\rightsquigarrow} v$ in $\mathcal{Q}$ by a set of edges $\{u \overset{L_1}{\rightsquigarrow} u_1, u_1 \overset{L_2}{\rightsquigarrow} u_2, \ldots, u_{\ell-1} \overset{L_\ell}{\rightsquigarrow} v\}$, where (i) the u_i's are fresh elements, (ii) $\ell \leq r$, and (iii) there is a sequence $s_0, \ldots, s_\ell$ of states of $\mathsf{A}(L)$ that satisfies the following: s_0 and s_ℓ are an initial and final state of $\mathsf{A}(L)$, respectively, and for each $1 \leq i \leq \ell$, it is the case that $L_i = \mathsf{E}(\mathsf{A}(L)(s_{i-1}, s_i))$. We denote by $\mathbb{SD}_r(\mathcal{Q})$ the set of all r-subdivisions of $\mathcal{Q}$.

For 2RGPs $\mathcal{Q}$ and $\mathcal{Q}'$, we say that $\mathcal{Q}'$ is a *quotient* of $\mathcal{Q}$ if $\mathcal{Q}'$ can be obtained from $\mathcal{Q}$ and a partition $V_1, \ldots, V_n$ of $\mathsf{Dom}(\mathcal{Q})$ by identifying all the elements in V_i with one fresh element v_i, for $1 \leq i \leq n$. For a set of 2RGPs $\mathbb{C}$, we denote by $\mathsf{Quot}(\mathbb{C})$ the set of all quotients of a 2RGP in $\mathbb{C}$.

Our algorithm constructs, given the 2RGP $\mathcal{P}$, the set $\mathfrak{R} = \mathsf{Quot}(\mathbb{SD}_r(\mathcal{P})) \cap \mathsf{TW}(2k+1)_{2\mathrm{rgp}}$, where $r = 2(k+1)|\mathcal{P}|^2$. It is routine to verify that the set $\mathfrak{R}$ is not empty and can be constructed in exponential time. By definition, $\mathfrak{R}$ satisfies condition (1) of the lemma. It remains to prove condition (2). We start by showing that $\bigcup_{\mathcal{R} \in \mathfrak{R}} \mathcal{R} \to \mathcal{G}$ implies $\mathcal{P} \to \mathcal{G}$, for each graph database $\mathcal{G}$. Suppose $\mathcal{R} \to \mathcal{G}$, for some $\mathcal{R} \in \mathfrak{R}$. In particular, $\mathcal{R} \in \mathsf{Quot}(\mathbb{SD}_r(\mathcal{P}))$, and therefore, there exists $\mathcal{R}' \in \mathbb{SD}_r(\mathcal{P})$ such that $\mathcal{R}$ is a quotient of $\mathcal{R}'$. By composing the renaming function that defines $\mathcal{R}$ from $\mathcal{R}'$ with the homomorphism that witnesses $\mathcal{R} \to \mathcal{G}$, we obtain that $\mathcal{R}' \to \mathcal{G}$. By definition of subdivisions, it follows that $\mathcal{P} \to \mathcal{G}$.

To show that $\mathcal{P} \to \mathcal{G}$ implies $\bigcup_{\mathcal{R} \in \mathfrak{R}} \mathcal{R} \to \mathcal{G}$, we need the following technical lemma:

Lemma IV.14. *Fix $k > 1$. Let $\mathcal{P}$ be a 2RGP and $\mathcal{G}$ be a graph database in $\mathsf{TW}(k)$. If $\mathcal{P} \to \mathcal{G}$, then $\mathcal{R} \to \mathcal{G}$, for some $\mathcal{R} \in \mathsf{Quot}(\mathbb{SD}_r(\mathcal{P})) \cap \mathsf{TW}(2k+1)_{2\mathrm{rgp}}$, where $r = 2(k+1)|\mathcal{P}|^2$.*

Proof Sketch: Let h be a homomorphism from $\mathcal{P}$ to $\mathcal{G}$ and let (T, β) be a tree decomposition of width k of the underlying undirected graph of $\mathcal{G}$. We define a set $\mathbf{V} \subseteq \mathsf{Dom}(\mathcal{G})$ as follows. Let $\mathbf{I} = \{h(u) \mid u \in \mathsf{Dom}(\mathcal{P})\}$. For a node $v \in \mathbf{I}$, let $t_v^{\downarrow}$ be the node in T that is the root of the subtree in T induced by $\{t \mid v \in \beta(t)\}$. Let $F = \{t_v^{\downarrow} \mid v \in \mathbf{I}\}$ and define F_{lca} to be the set of nodes of T that contains F and all the least common ancestors in T of every pair of nodes in F. Let T_{lca} be the (rooted) tree induced by the tree structure of T over the nodes in F_{lca}, and let β_{lca} be the restriction of β to the nodes of T_{lca}. Then the set $\mathbf{V}$ is defined as $\mathbf{V} = \bigcup_{t \in T_{lca}} \beta_{lca}(t)$. Note that $\mathbf{I} \subseteq \mathbf{V}$. Also, it can be verified that $|\mathbf{V}| \leq 2(k+1)|\mathsf{Dom}(\mathcal{P})|$.

Suppose $\mathcal{P}$ is of the form $\{u_i \overset{L_i}{\rightsquigarrow} v_i \mid 1 \leq i \leq n\}$. Let $i \in \{1, \ldots, n\}$. Since h is a homomorphism, there is a path ρ_i in the completion $\mathcal{G}^{\pm}$ of $\mathcal{G}$ from $h(u_i)$ to $h(v_i)$, whose label matches L_i. Since $h(u_i), h(v_i) \in \mathbf{V}$, we can decompose

ρ_i into a sequence of paths $\rho_i^1, \ldots, \rho_i^{m_i}$ such that (i) its concatenation is precisely ρ, and (ii) for each $1 \leq j \leq m_i$, the endpoints of ρ_i^j are in $\mathbf{V}$, while all the internal nodes of ρ_i^j are not in $\mathbf{V}$. Using pumping arguments, we can choose m_i to be at most $|\mathbf{V}| \cdot |L_i| \leq 2(k+1)|\mathcal{P}|^2 = r$. Let $w_i^0, w_i^1, \ldots, w_i^{m_i}$ be the sequence of nodes from $\mathbf{V}$ such that the endpoints of ρ_i^j are w_i^{j-1} and w_i^j, for each $1 \leq j \leq m_i$. In particular, $w_i^0 = h(u_i)$ and $w_i^{m_i} = h(v_i)$. Now let π be an accepting run of $A(L_i)$ on the word $\lambda(\rho_i)$, i.e., the label of ρ_i. Let $s_i^0, s_i^1 \ldots, s_i^{m_i}$ be the sequence of states of $\mathsf{A}(L_i)$ such that $\lambda(\rho_i^j)$ goes from s_{j-1} to s_j in the run π. In particular, s_0 and s_{m_i} are the initial and final states in the run π, respectively.

Now we are ready to define the 2RGP $\mathcal{R}$. The domain of $\mathcal{R}$ is $\mathsf{Dom}(\mathcal{R}) = \mathbf{V}$. For each $1 \leq i \leq n$, let $w_i^0, w_i^1, \ldots, w_i^{m_i}$ and $s_i^0, s_i^1, \ldots, s_i^{m_i}$ be the sequences defined above. For each $1 \leq j \leq m_i$, we have an edge in $\mathcal{R}$ of the form $w_i^{j-1} \overset{L_i^j}{\rightsquigarrow} w_i^j$, where $L_i^j = \mathsf{E}(\mathsf{A}(L_i)(s_i^{j-1}, s_i^j))$.

Notice that $\mathcal{R} \to \mathcal{G}$ via the identity homomorphism. Also, observe that $\mathcal{R} \in \mathsf{Quot}(\mathbb{SD}_r(\mathcal{P}))$. Indeed, by construction, we have that $\mathcal{R} \in \mathsf{Quot}(\mathbb{SD}_{r'}(\mathcal{P}))$, where $r' = max\{m_i \mid 1 \leq i \leq n\}$. As noted above, $m_i \leq r$, for each $1 \leq i \leq n$. Finally, from the pair (T_{lca}, β_{lca}) one can define a tree decomposition (T', β') for $\mathcal{R}$ of width at most $2k+1$. The idea is to show that whenever $w \overset{L}{\rightsquigarrow} w'$ is an edge in $\mathcal{R}$, then either $\{w, w'\} \subseteq \beta_{lca}(t)$, for some $t \in T_{lca}$, or $w \in \beta_{lca}(t_1)$ and $w' \in \beta_{lca}(t_2)$ for an edge $\{t_1, t_2\}$ in T_{lca}. With this property at hand, (T', β') is obtained from (T_{lca}, β_{lca}) by adding for each edge $\{t_1, t_2\}$ a node $t_{1,2}$ "between" t_1 and t_2 with label $\beta'(t_{1,2}) = \beta_{lca}(t_1) \cup \beta_{lca}(t_2)$. As it turns out, (T', β') is actually a tree decomposition of width at most $2k+1$ and then $\mathcal{R} \in \mathsf{TW}(2k+1)_{2\mathrm{rgp}}$ as required. □

To conclude the lemma we need the notion of *expansions* (also known as *canonical databases* in [17], [23]) of a 2RGP. The idea is that an expansion is a graph database obtained from a 2RGP $\mathcal{Q}$ by replacing each edge $u \overset{L}{\rightsquigarrow} v$ by a fresh path from u to v with a label satisfying L. As it turns out, a 2RGP $\mathcal{Q}$ is "equivalent" to its (potentially infinite) set of expansions, i.e., (†) for each graph database $\mathcal{G}$, we have that $\mathcal{Q} \to \mathcal{G}$ iff $\mathcal{H} \to \mathcal{G}$, for some expansion $\mathcal{H}$ of $\mathcal{Q}$. Moreover, (‡) if $\mathcal{Q} \in \mathsf{TW}(k)_{2\mathrm{rgp}}$ and $\mathcal{H}$ is an expansion of $\mathcal{Q}$, then $\mathcal{H} \in \mathsf{TW}(k)$. Here our assumption $k > 1$ is crucial: a 2RGP of the form $\{u \overset{L}{\rightsquigarrow} v, u \overset{L'}{\rightsquigarrow} v\}$ belongs to $\mathsf{TW}(1)_{2\mathrm{rgp}}$ but its expansions may belong to $\mathsf{TW}(2) \setminus \mathsf{TW}(1)$.

We prove that $\mathcal{P} \to \mathcal{G}$ implies $\bigcup_{\mathcal{R} \in \mathfrak{R}} \mathcal{R} \to \mathcal{G}$, for each graph database $\mathcal{G}$. Assume that $\mathcal{P} \to \mathcal{G}$. Let $\mathcal{P}'$ be a 2RGP in $\mathsf{TW}(k)_{2\mathrm{rgp}}$ which is a witness for $\mathcal{P} \in \mathsf{TW}(k)_{\equiv_{2\mathrm{rgp}}}$. Then $\mathcal{P} \to \mathcal{G}$, since $\mathcal{P}' \to \mathcal{G}$. By (†), $\mathcal{H}' \to \mathcal{G}$, for some expansion $\mathcal{H}'$ of $\mathcal{P}'$. Also by (†), $\mathcal{P}' \to \mathcal{H}'$ and then $\mathcal{P} \to \mathcal{H}'$. By (‡), we know that $\mathcal{H}' \in \mathsf{TW}(k)$. We apply Lemma IV.14, and obtain a 2RGP $\mathcal{R} \in \mathsf{Quot}(\mathbb{SD}_r(\mathcal{P})) \cap \mathsf{TW}(2k+1)_{2\mathrm{rgp}}$, for $r = 2(k+1)|\mathcal{P}|^2$, such that $\mathcal{R} \to \mathcal{H}'$. Clearly, $\mathcal{R} \in \mathfrak{R}$. Since $\mathcal{H}' \to \mathcal{G}$, it follows that $\mathcal{R} \to \mathcal{G}$. Thus, $\bigcup_{\mathcal{R} \in \mathfrak{R}} \mathcal{R} \to \mathcal{G}$. □

Now we explain how Theorem IV.11 follows from Lemma IV.13. We are given a 2RGP $\mathcal{P}$ in $\mathsf{TW}(k)_{\equiv_{2\mathrm{rgp}}}$ and a graph database $\mathcal{G}$, and we want to check if $\mathcal{P} \to \mathcal{G}$. We then apply Lemma IV.13 to construct in exponential time the set $\mathfrak{R}$. From condition (2), checking if $\mathcal{P} \to \mathcal{G}$ boils down then to check whether $\bigcup_{\mathcal{R} \in \mathfrak{R}} \mathcal{R} \to \mathcal{G}$, i.e., $\mathcal{R} \to \mathcal{G}$ for some $\mathcal{R} \in \mathfrak{R}$. We concentrate on the latter. Recall that $\mathfrak{R}$ consists of at most exponentially many 2RGPs $\mathcal{R}$ of exponential size, and each such 2RGP $\mathcal{R}$ is in $\mathsf{TW}(2k+1)_{2\mathrm{rgp}}$ (from condition (1)). It follows then from Theorem IV.3 that checking if $\bigcup_{\mathcal{R} \in \mathfrak{R}} \mathcal{R} \to \mathcal{G}$ is feasible in time $\sum_{\mathcal{R} \in \mathfrak{R}} (|\mathcal{R}| + |\mathcal{G}|)^{O(k)}$, which is $O\big(|\mathcal{G}|^{t(k)} \cdot 2^{s(|\mathcal{P}|)}\big)$ for $t : \mathbb{N} \to \mathbb{N}$ a linear function and $s : \mathbb{N} \to \mathbb{N}$ a polynomial.

As already noticed in the proof of Lemma IV.13, we have previously established a small-witness property – similar to the one stated in Lemma IV.13 – for the case $k = 1$ [23]. Interestingly, in that case, it is possible to construct the "witness" $\mathfrak{R}$ in such a way that each 2RPQ $\mathcal{R} \in \mathfrak{R}$ is in $\mathsf{TW}(1)_{2\mathrm{rgp}}$. Extending such techniques to an arbitrary $k > 1$ requires new insights. In particular, we no longer obtain in Lemma IV.13 a witness $\mathfrak{R}$ composed exclusively of 2RGPs in $\mathsf{TW}(k)_{2\mathrm{rgp}}$. Hence we need to be more flexible and allow $\mathfrak{R}$ to contain 2RGPs in $\mathsf{TW}(2k+1)_{2\mathrm{rgp}}$.

V. Equivalence based on unions of C2RPQs

The notion of homomorphism for sets of 2RGPs used in Lemma IV.13 naturally leads to an extended homomorphism problem for a class $\mathbb{C}$ of sets of 2RGPs, defined as follows: Given a set $\mathfrak{P} \in \mathbb{C}$ and a graph database $\mathcal{G}$, is it the case that $\bigcup_{\mathcal{P} \in \mathfrak{P}} \mathcal{P} \to \mathcal{G}$? We abuse notation and keep writing $\mathsf{2RGPHom}(\mathbb{C}, -)$ for this problem.

The notion of treewidth extends to unions of 2RGPs, and also ensures the tractability of $\mathsf{2RGPHom}(\mathbb{C}, -)$. Formally, we write $\mathsf{UTW}(k)_{2\mathrm{rgp}}$ for the class of sets of 2RGPs $\mathfrak{P}$ such that each $\mathcal{P} \in \mathfrak{P}$ is in $\mathsf{TW}(k)_{2\mathrm{rgp}}$. We immediately obtain from Theorem IV.3 that $\mathsf{2RGPHom}(\mathsf{UTW}(k)_{2\mathrm{rgp}}, -)$ is tractable.

As 2RGPs are associated with C2RPQs, sets of 2RGPs can be associated with *unions of C2RPQs*, or UC2RPQs. A UC2RPQ Φ over Σ is an expression of the form $\bigcup_{1 \leq i \leq m} \phi_i$, where each ϕ_i is a C2RPQ over Σ. If Φ is a Boolean UC2RPQ and $\mathcal{G}$ is a graph database, we have that $\mathcal{G} \models \Phi$ iff $\mathcal{G} \models \phi_i$ for some $1 \leq i \leq m$. It is clear then that if $\{\mathcal{P}_1, \ldots, \mathcal{P}_n\}$ is a set of 2RGPs, then for each graph database $\mathcal{G}$ it is the case that: $\bigcup_{1 \leq i \leq n} \mathcal{P}_i \to \mathcal{G}$ iff $\mathcal{G} \models \bigcup_{1 \leq i \leq n} \phi_{\mathcal{P}_i}$. Correspondingly, if $\bigcup_{1 \leq i \leq n} \phi_i$ is a UC2RPQ, then for each graph database $\mathcal{G}$ it is the case that: $\mathcal{G} \models \bigcup_{1 \leq i \leq n} \phi_i$ iff $\bigcup_{1 \leq i \leq n} \mathcal{P}_{\phi_i} \to \mathcal{G}$. Therefore, the evaluation problem for the class $\mathsf{UTW}(k)^{\Phi}_{2\mathrm{rgp}}$, defined as the UC2RPQs $\Phi = \bigcup_{1 \leq i \leq n} \phi_i$ such that $\{\mathcal{P}_{\phi_1}, \ldots, \mathcal{P}_{\phi_n}\} \in \mathsf{UTW}(k)_{2\mathrm{rgp}}$, is tractable.

The connection between sets of 2RGPs and UC2RPQs naturally leads to a notion of bounded treewidth modulo equivalence for sets of 2RGPs based on the equivalence of their associated UC2RPQs. We formalize this next. Let us write $\Phi \equiv \Phi'$, for UC2RPQs Φ, Φ', if for every $\mathcal{G}$ it is the case that $\mathcal{G} \models \Phi \Leftrightarrow \mathcal{G} \models \Phi'$. We then define $\mathsf{UTW}(k)^{\Phi}_{\equiv_{2\mathrm{rgp}}}$, for $k \geq 1$, as the class of Boolean UC2RPQs Φ such that there is a UC2RPQ Φ' with $\Phi \equiv \Phi'$ and $\Phi' \in \mathsf{UTW}(k)^{\Phi}_{2\mathrm{rgp}}$.

Analogously, we define $\mathsf{UTW}(k)_{\equiv_{2\mathrm{rgp}}}$ as the class of sets of 2RGPs whose associated UC2RPQ Φ is in $\mathsf{UTW}(k)^{\Phi}_{\equiv_{2\mathrm{rgp}}}$.

The good behavior of $\mathsf{TW}(k)_{\equiv_{2\mathrm{rgp}}}$ for $\mathsf{2RGPHom}(\mathbb{C}, -)$ established in Theorem IV.11 extends to $\mathsf{UTW}(k)_{\equiv_{2\mathrm{rgp}}}$:

Theorem V.1. *Fix $k \geq 1$. Then $\mathsf{2RGPHom}(\mathsf{UTW}(k)_{\equiv_{2\mathrm{rgp}}}, -)$ is feasible in time $O\big(|\mathcal{G}|^{t(k)} \cdot 2^{s(|\mathfrak{P}|)}\big)$, for $t : \mathbb{N} \to \mathbb{N}$ a linear function and $s : \mathbb{N} \to \mathbb{N}$ a polynomial.*

The reason that explains this good behavior is that Lemma IV.13 continues to hold when the input is given as a set of 2RGPs. More formally, for each fixed $k \geq 1$ there is a single-exponential time algorithm that on input a set $\mathfrak{P}$ of 2RGPs in $\mathsf{UTW}(k)_{\equiv_{2\mathrm{rgp}}}$, computes a set $\mathfrak{R} \in \mathsf{UTW}(2k{+}1)_{2\mathrm{rgp}}$ such that for each graph database $\mathcal{G}$: $\bigcup_{\mathcal{P}\in\mathfrak{P}} \mathcal{P} \to \mathcal{G}$ iff $\bigcup_{\mathcal{R}\in\mathfrak{R}} \mathcal{R} \to \mathcal{G}$.

As a corollary of Theorem V.1, we obtain the following:

Corollary V.2. *Fix $k \geq 1$. Evaluation for UC2RPQs in $\mathsf{UTW}(k)^{\Phi}_{\equiv_{2\mathrm{rgp}}}$ is feasible in time $O(|\mathcal{G}|^{t(k)} \cdot 2^{s(|\Phi|)})$, for $t : \mathbb{N} \to \mathbb{N}$ a linear function and $s : \mathbb{N} \to \mathbb{N}$ a polynomial.*

This generalizes a result in [23] that establishes the fixed-parameter tractability of evaluation for the class $\mathsf{UTW}(1)^{\Phi}_{\equiv_{2\mathrm{rgp}}}$ of semantically acyclic UC2RPQs.

Optimality of the previous results: The fixed-parameter tractability of $\mathsf{2RGPHom}(\mathsf{UTW}(k)_{\equiv_{2\mathrm{rgp}}}, -)$ established in Theorem V.1 is, in a sense, optimal, since the problem is also NP-complete. This closes an open problem from [23]:

Theorem V.3. *For each fixed $k \geq 1$ the problem $\mathsf{2RGPHom}(\mathsf{UTW}(k)_{\equiv_{2\mathrm{rgp}}}, -)$ is* NP*-complete. Consequently, evaluation for UC2RPQs in $\mathsf{UTW}(k)^{\Phi}_{\equiv_{2\mathrm{rgp}}}$ is* NP*-complete.*

Proof Sketch: Our proof is inspired by a proof in [15] that shows that evaluation for $\exists^{+}\mathrm{FO}$ is NP-complete, even for sentences that are equivalent to an $\exists^{+}\mathrm{FO}$ sentence that uses a fixed number of variables. Our construction is different, though, as UC2RPQs are weak in their ability to nest subqueries, while $\exists^{+}\mathrm{FO}$ sentences have unlimited power to do this. This complicates things for us, specially at the moment of constructing in our reduction a UC2RPQ in $\mathsf{UTW}(k)^{\Phi}_{\equiv_{2\mathrm{rgp}}}$.

We reduce from the DIRECTED HAMILTONIAN PATH problem: Given a directed graph G, does G contain a directed hamiltonian path? Let G be a directed graph with vertex set $\{v_1, \ldots, v_n\}$. We construct in polynomial time a graph database $\mathcal{H}_G$ and a set $\mathfrak{P}_G$ of 2RGPs in $\mathsf{UTW}(1)_{\equiv_{2\mathrm{rgp}}}$, such that G has a directed hamiltonian path iff $\bigcup_{\mathcal{P}\in\mathfrak{P}_G} \mathcal{P} \to \mathcal{H}_G$.

Let $\mathcal{G}_1$ be the graph database over alphabet $\{a, \$_1, \ldots, \$_n\}$ whose edges are $\{v_i \overset{\$_i}{\rightsquigarrow} v_i \mid 1 \leq i \leq n\} \cup \{v_i \overset{a}{\rightsquigarrow} v_j \mid (v_i, v_j)$ is an edge in $G\}$. Let also $\mathcal{G}_2$ be the graph database whose edges are $\{s_i \overset{\$_j}{\rightsquigarrow} s_i \mid 1 \leq i, j \leq n\} \cup \{s_i \overset{a}{\rightsquigarrow} s_{i+1} \mid 1 \leq i \leq n-1\}$. We define $\mathcal{H}_G := \mathcal{G}_1 \otimes \mathcal{G}_2$, where $\otimes$ denotes the usual direct product between relational structures.

We now define $\mathfrak{P}_G$. Let $\mathcal{R}$ be the 2RGP whose edges are $\{u_i \overset{\$_i}{\rightsquigarrow} u_i,\ u_i \overset{F_n}{\rightsquigarrow} u_j \mid 1 \leq i, j \leq n$ and $i \neq j\} \cup \{\$_i(u_i, u_i) \mid 1 \leq i \leq n\}$, where F_n is the regular expression:

$$F_n = (a + a^2 + \cdots + a^{n-1}) + (a^{-} + (a^{-})^2 + \cdots + (a^{-})^{n-1}).$$

Also, for each $1 \leq i, j \leq n$ and $p, q \geq 1$, let $\mathcal{P}^{i,j}_{p,q} := \{u \overset{\$_i}{\rightsquigarrow} u,\ v \overset{\$_j}{\rightsquigarrow} v,\ u \overset{a^p}{\rightsquigarrow} v,\ u \overset{a^q}{\rightsquigarrow} v\}$. We then define a set of 2RGPs:

$$\mathfrak{R} = \bigcup\{\mathcal{P}^{i,j}_{p,q} \mid, 1 \leq i, j \leq n,\ 1 \leq p, q \leq n^2,\ i \neq j,\ p \neq q\}.$$

We finally define $\mathfrak{P}_G := \mathcal{R} \cup \mathfrak{R}$. Clearly, $\mathcal{H}_G$ and $\mathfrak{P}_G$ can be constructed in polynomial time from G. Next we show the correctness of the reduction; i.e., (a) $\mathfrak{P}_G \in \mathsf{UTW}(1)_{\equiv_{2\mathrm{rgp}}}$, and (b) G has a directed hamiltonian path iff $\bigcup_{\mathcal{P}\in\mathfrak{P}_G} \mathcal{P} \to \mathcal{H}_G$.

We start by proving (a). Let us consider a particular set of graph databases, which we call *good*. In particular, each such good graph database is associated with a permutation $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ in the following way: the edges of the good graph database $\mathcal{G}_\pi$ associated with permutation π correspond precisely to the set $\{v_i \overset{\$_i}{\rightsquigarrow} v_i,\ v_{\pi(i)} \overset{a}{\rightsquigarrow} v_{\pi(i+1)} \mid 1 \leq i \leq n\}$. It is easy to see that $\mathcal{R} \to \mathcal{G}_\pi$.

Let us define now a set of 2RGPs $\mathfrak{P}'$ that contains precisely the good graph databases, as well as all 2RGPs in $\mathfrak{R}$. It is easy to see that each 2RGP in $\mathfrak{P}'$ is in $\mathsf{TW}(1)_{2\mathrm{rgp}}$, and hence $\mathfrak{P}' \in \mathsf{UTW}(1)_{2\mathrm{rgp}}$. We claim the following:

Claim V.4. *$\Phi_{\mathfrak{P}_G} \equiv \Phi_{\mathfrak{P}'}$, where $\Phi_{\mathfrak{P}_G}$ and $\Phi_{\mathfrak{P}'}$ are the UC2RPQs associated with $\mathfrak{P}_G$ and $\mathfrak{P}'$, respectively.*

From Claim V.4 we obtain that $\mathfrak{P}_G \in \mathsf{UTW}(1)_{\equiv_{2\mathrm{rgp}}}$, as desired. We provide now an intuitive explanation of why the claim holds. Assume first that it is the case that $\mathcal{G} \models \Phi_{\mathfrak{P}'}$. If $\mathcal{G} \models \phi_{\mathcal{P}}$ for $\mathcal{P} \in \mathfrak{R}$, then it is also the case that $G \models \Phi_{\mathfrak{P}_G}$ (since $\mathfrak{R} \subseteq \mathfrak{P}_G$). Suppose then that $\mathcal{G} \models \phi_{\mathcal{G}_\pi}$ for $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$. Then it can be proved that $\mathcal{G} \models \mathcal{R}$ using the fact that $\mathcal{R} \to \mathcal{G}_\pi$. Hence $G \models \Phi_{\mathfrak{P}_G}$ since $\mathcal{R} \in \mathfrak{P}_G$. Assume, on the other hand, that $\mathcal{G} \models \Phi_{\mathfrak{P}_G}$. If $\mathcal{G} \models \phi_{\mathcal{P}}$ for $\mathcal{P} \in \mathfrak{R}$, we proceed as before. Suppose then that $\mathcal{G} \models \mathcal{R}$. Let us consider the "image" of $\mathcal{R}$ in $\mathcal{G}$. If such an image corresponds to a good graph database, then $\mathcal{G} \models \phi_{\mathcal{G}_\pi}$ for some $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$. If not, then it is possible to prove that there exist two elements u and v in such an image such that there are two paths from u to v labeled a^p and a^q, respectively, where $1 \leq p, q \leq n^2$ and $p \neq q$. Hence $\mathcal{G} \models \phi_{\mathcal{P}}$ for some $\mathcal{P} \in \mathfrak{R}$, which implies that $\mathcal{G} \models \Phi_{\mathfrak{P}'}$.

We now prove (b). Assume first that G has a hamiltonian path given by a permutation π of $\{1, \ldots, n\}$. It follows that $\mathcal{G}_\pi \to \mathcal{G}_1$. Notice that also $\mathcal{G}_\pi \to \mathcal{G}_2$ (in fact, this holds for any permutation π). Then $\mathcal{G}_\pi \to \mathcal{G}_1 \otimes \mathcal{G}_2$, i.e., $\mathcal{G}_\pi \to \mathcal{H}_G$, and, therefore, $\mathfrak{P}_G \to \mathcal{H}_G$. Assume, on the other hand, that $\mathfrak{P}_G \to \mathcal{H}_G$. Then for some 2RGP $\mathcal{P} \in \mathfrak{P}_G$ it is the case that $\mathcal{P} \to \mathcal{H}_G$. In particular, $\mathcal{P} \to \mathcal{G}_1$ and $\mathcal{P} \to \mathcal{G}_2$. By construction, if $\mathcal{P} \to \mathcal{G}_2$ then $\mathcal{P} \notin \mathfrak{R}$. It follows that $\mathcal{P} = \mathcal{G}_\pi$ for some permutation π of $\{1, \ldots, n\}$. We conclude that the directed path defined by π is a hamiltonian path in G. (It is worth noticing that it is also the case that G has a directed hamiltonian path iff $\mathcal{R} \to \mathcal{H}_G$. However, $\mathcal{R}$ is not in $\mathsf{UTW}(1)_{\equiv_{2\mathrm{rgp}}}$. The role of the 2RGPs is thus to ensure that $\mathfrak{P}_G = \mathcal{R} \cup \mathfrak{R}$ is in $\mathsf{UTW}(1)_{\equiv_{2\mathrm{rgp}}}$). □

Theorem V.3 states that the notion of fixed-parameter tractability properly extends tractability (under usual complexity theoretical assumptions) for the problem $\mathsf{2RGPHom}(\mathbb{C}, -)$

under classes $\mathbb{C}$ of sets of 2RGPs. This establishes a stark difference with the usual problem $\mathsf{Hom}(\mathbb{C}, -)$ for classes $\mathbb{C}$ of sets of graph databases; i.e., given a set $\mathfrak{G}$ of graph databases and a graph database $\mathcal{H}$, is it the case that for some $\mathcal{G} \in \mathfrak{G}$ we have that $\mathcal{G} \to \mathcal{H}$? In fact, by combining results from [12] and [15] we obtain that the notion of fixed-parameter tractability coincides with tractability in such context (again under usual complexity theoretical assumptions).

Decidability of the notion: We proved in [23] that the problem of checking if a set $\mathfrak{P}$ of 2RGPs is in $\mathsf{UTW}(1)_{\equiv_{2\mathrm{rgp}}}$ is decidable, and actually EXPSPACE-complete. We do not know if the decidability extends to $\mathsf{UTW}(k)_{\equiv_{2\mathrm{rgp}}}$, for $k > 1$. The difference resides precisely on our small-witness property. As mentioned before, in the case $k = 1$ we can construct a "witness" $\mathfrak{R}$ that consists exclusively of 2RGPs in $\mathsf{TW}(1)_{2\mathrm{rgp}}$. Such set $\mathfrak{R}$ can then be used to witness the fact that $\mathfrak{P} \in \mathsf{UTW}(1)_{\equiv_{2\mathrm{rgp}}}$. For $k > 1$, on the other hand, we are only able to construct a "witness" $\mathfrak{R} \in \mathsf{UTW}(2k+1)_{2\mathrm{rgp}}$. This does not suffice to ensure the decidability of the notion.

VI. Conclusions and open problems

We have studied the homomorphism problem for 2RGPs. In particular, its non-uniform version is computationally harder than for the usual homomorphism notion. While for RGPs establishing a dichotomy seems difficult, for 2RGPs the problem might be easier since the intractability conditions are very general. We have also considerably expanded the frontier of efficient solvability for $\mathsf{2RGPHom}(\mathbb{C}, -)$, showing how it relates to the notion of bounded treewidth modulo equivalence.

Many problems still remain open. We list some of them:

1) Prove a dichotomy for the non-uniform problems $\mathsf{2RGPHom}(\mathcal{G})$ and $\mathsf{RGPHom}(\mathcal{G})$.
2) We know that the problem $\mathsf{2RGPHom}(\mathsf{TW}(k)_{\equiv_{2\mathrm{rgp}}}, -)$ is fixed-parameter tractable. Is this problem also NP-complete? Recall that this is known to be the case for $\mathsf{2RGPHom}(\mathsf{UTW}(k)_{\equiv_{2\mathrm{rgp}}}, -)$, for each $k \geq 1$.
3) Study the decidability status of $\mathsf{UTW}(k)_{\equiv_{2\mathrm{rgp}}}$ for $k > 1$: Given a set $\mathfrak{P}$ of 2RGPs, is $\mathfrak{P} \in \mathsf{UTW}(k)_{\equiv_{2\mathrm{rgp}}}$?
4) Understand the efficiency boundary for $\mathsf{2RGPHom}(\mathbb{C}, -)$. In particular, does bounded treewidth modulo logical equivalence exhaust fixed-parameter tractability for $\mathsf{2RGPHom}(\mathbb{C}, -)$?

Acknowledgment

This work was done in part while Romero was visiting the Simons Institute for the Theory of Computing. Barceló is funded by the Millennium Nucleus Center for Semantic Web Research under Grant NC120004.

References

[1] R. Dechter, *Constraint processing*. Elsevier Morgan Kaufmann, 2003.
[2] S. Abiteboul, R. Hull, and V. Vianu, *Foundations of Databases*. Addison-Wesley, 1995.
[3] T. Feder and M. Y. Vardi, "The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory," *SIAM J. Comput.*, vol. 28, no. 1, pp. 57–104, 1998.
[4] A. A. Bulatov, A. A. Krokhin, and P. Jeavons, "Constraint satisfaction problems and finite algebras," in *ICALP*, 2000, pp. 272–282.
[5] A. A. Bulatov, P. Jeavons, and A. A. Krokhin, "Classifying the complexity of constraints using finite algebras," *SIAM J. Comput.*, vol. 34, no. 3, pp. 720–742, 2005.
[6] M. Maróti and R. McKenzie, "Existence theorems for weakly symmetric operations," *Algebra Universalis*, vol. 59, no. 3-4, pp. 463–489, 2008.
[7] A. Rafiey, J. Kinne, and T. Feder, "Dichotomy for digraph homomorphism problems," *CoRR*, 2017. [Online]. Available: http://arxiv.org/abs/1701.02409
[8] A. A. Bulatov, "A dichotomy theorem for nonuniform csps," *CoRR*, 2017. [Online]. Available: http://arxiv.org/abs/1703.03021
[9] D. Zhuk, "The proof of csp dichotomy conjecture," *CoRR*, 2017. [Online]. Available: https://arxiv.org/abs/1704.01914
[10] C. Chekuri and A. Rajaraman, "Conjunctive query containment revisited," *Theor. Comput. Sci.*, vol. 239, no. 2, pp. 211–229, 2000.
[11] V. Dalmau, P. G. Kolaitis, and M. Vardi, "Constraint satisfaction, bounded treewidth, and finite-variable logics," in *CP*, 2002, pp. 310–326.
[12] M. Grohe, "The complexity of homomorphism and constraint satisfaction problems seen from the other side," *J. ACM*, vol. 54, no. 1, 2007.
[13] B. Martin, "Dichotomies and duality in first-order model checking problems," *CoRR*, vol. abs/cs/0609022, 2006.
[14] M. Hermann and F. Richoux, "On the computational complexity of monotone constraint satisfaction problems," in *WALCOM*, 2009, pp. 286–297.
[15] H. Chen, "On the complexity of existential positive queries," *ACM Trans. Comput. Log.*, vol. 15, no. 1, p. 9, 2014.
[16] P. Barceló, "Querying graph databases," in *PODS*, 2013, pp. 175–188.
[17] D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Y. Vardi, "Containment of conjunctive regular path queries with inverse," in *KR*, 2000, pp. 176–185.
[18] "SPARQL 1.1 Query Language," https://www.w3.org/TR/sparql11-query/. W3C Recommendation 21 March 2013.
[19] O. van Rest, S. Hong, J. Kim, X. Meng, and H. Chafi, "PGQL: a property graph query language," in *GRADES*, 2016, p. 7.
[20] R. Angles, M. Arenas, P. Barceló, A. Hogan, J. L. Reutter, and D. Vrgoc, "Foundations of modern graph query languages," *CoRR*, vol. abs/1610.06264, 2016.
[21] D. D. Freydenberger and N. Schweikardt, "Expressiveness and static analysis of extended conjunctive regular path queries," *J. Comput. Syst. Sci.*, vol. 79, no. 6, pp. 892–909, 2013.
[22] D. Figueira and L. Libkin, "Path logics for querying graphs: Combining expressiveness and efficiency," in *LICS*, 2015, pp. 329–340.
[23] P. Barceló, M. Romero, and M. Y. Vardi, "Semantic acyclicity on graph databases," *SIAM J. Comput.*, vol. 45, no. 4, pp. 1339–1376, 2016.
[24] D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Y. Vardi, "Rewriting of regular expressions and regular path queries," *J. Comput. Syst. Sci.*, vol. 64, no. 3, pp. 443–465, 2002.
[25] S. Fortune, J. E. Hopcroft, and J. Wyllie, "The directed subgraph homeomorphism problem," *TCS*, vol. 10, pp. 111–121, 1980.
[26] P. G. Kolaitis and M. Y. Vardi, "On the expressive power of datalog: Tools and a case study," *J. Comput. Syst. Sci.*, vol. 51, no. 1, pp. 110–134, 1995.
[27] J. Bulin, D. Delic, M. Jackson, and T. Niven, "On the reduction of the CSP dichotomy conjecture to digraphs," in *CP*, 2013, pp. 184–199.
[28] P. Barceló and M. Romero, "The complexity of reverse engineering problems for conjunctive queries," in *ICDT*, 2017.
[29] C. H. Papadimitriou and M. Yannakakis, "On the complexity of database queries," *J. Comput. Syst. Sci.*, vol. 58, no. 3, pp. 407–427, 1999.
[30] P. G. Kolaitis and M. Y. Vardi, "Conjunctive-query containment and constraint satisfaction," *J. Comput. Syst. Sci.*, vol. 61, no. 2, pp. 302–332, 2000.
[31] P. Barceló, L. Libkin, and J. L. Reutter, "Querying regular graph patterns," *J. ACM*, vol. 61, no. 1, pp. 8:1–8:54, 2014.
[32] P. Hell and J. Nešetřil, *Graphs and homomorphisms*. Oxford University Press, 2004.
[33] R. Willard, "Testing expressibility is hard," in *CP*, 2010, pp. 9–23.
[34] R. Diestel, *Graph Theory, 4th Edition*, ser. Graduate texts in mathematics. Springer, 2012, vol. 173.
[35] D. Kozen, "Lower bounds for natural proof systems," in *FOCS*, 1977, pp. 254–266.
[36] A. K. Chandra and P. M. Merlin, "Optimal implementation of conjunctive queries in relational data bases," in *STOC*, 1977, pp. 77–90.