Constraint Satisfaction with Counting Quantifiers 2

Barnaby Martin[1⋆] and Juraj Stacho[2]

[1] School of Science and Technology, Middlesex University
The Burroughs, Hendon, London NW4 4BT, U.K.
[2] IEOR Department, Columbia University,
500 West 120th Street, New York, NY 10027, United States

Abstract. We study *constraint satisfaction problems* (CSPs) in the presence of counting quantifiers $\exists^{\geq j}$, asserting the existence of j distinct witnesses for the variable in question. As a continuation of our previous (CSR 2012) paper [11], we focus on the complexity of undirected graph templates. As our main contribution, we settle the two principal open questions proposed in [11]. Firstly, we complete the classification of clique templates by proving a full trichotomy for all possible combinations of counting quantifiers and clique sizes, placing each case either in **P**, **NP**-complete or **PSPACE**-complete. This involves resolution of the cases in which we have the single quantifier $\exists^{\geq j}$ on the clique $\mathbb{K}_{2j}$. Secondly, we confirm a conjecture from [11], which proposes a full dichotomy for $\exists$ and $\exists^{\geq 2}$ on all finite undirected graphs. The main thrust of this second result is the solution of the complexity for the infinite path which we prove is a polynomial-time solvable problem. By adapting the algorithm for the infinite path we are then able to solve the problem for finite paths, and then trees and forests. Thus as a corollary to this work, combining with the other cases from [11], we obtain a full dichotomy for $\exists$ and $\exists^{\geq 2}$ quantifiers on finite graphs, each such problem being either in **P** or **NP**-hard. Finally, we persevere with the work of [11] in exploring cases in which there is dichotomy between **P** and **PSPACE**-complete, and contrast this with situations in which the intermediate **NP**-completeness may appear.

1 Introduction

The *constraint satisfaction problem* CSP($\mathcal{B}$), much studied in artificial intelligence, is known to admit several equivalent formulations, two of the best known of which are the query evaluation of primitive positive (pp) sentences – those involving only existential quantification and conjunction – on $\mathcal{B}$, and the homomorphism problem to $\mathcal{B}$ (see, e.g., [9]). The problem CSP($\mathcal{B}$) is **NP**-complete in general, and a great deal of effort has been expended in classifying its complexity for certain restricted cases. Notably it is conjectured [7, 4] that for all fixed $\mathcal{B}$, the problem CSP($\mathcal{B}$) is in **P** or **NP**-complete. While this has not been settled in general, a number of partial results are known – e.g. over structures of size

⋆ The author was supported by EPSRC grant EP/L005654/1.

at most three [13, 3] and over smooth digraphs [8, 1]. A popular generalization of the CSP involves considering the query evaluation problem for *positive Horn* logic – involving only the two quantifiers, $\exists$ and $\forall$, together with conjunction. The resulting *quantified constraint satisfaction problems* QCSP($\mathcal{B}$) allow for a broader class, used in artificial intelligence to capture non-monotonic reasoning, whose complexities rise to **PSPACE**-completeness.

In this paper, we continue the project begun in [11] to study counting quantifiers of the form $\exists^{\geq j}$, which allow one to assert the existence of at least j elements such that the ensuing property holds. Thus on a structure $\mathcal{B}$ with domain of size n, the quantifiers $\exists^{\geq 1}$ and $\exists^{\geq n}$ are precisely $\exists$ and $\forall$, respectively.

We study variants of CSP($\mathcal{B}$) in which the input sentence to be evaluated on $\mathcal{B}$ (of size $|B|$) remains positive conjunctive in its quantifier-free part, but is quantified by various counting quantifiers.

For $X \subseteq \{1, \ldots, |B|\}$, $X \neq \emptyset$, the X-CSP($\mathcal{B}$) takes as input a sentence given by a conjunction of atoms quantified by quantifiers of the form $\exists^{\geq j}$ for $j \in X$. It then asks whether this sentence is true on $\mathcal{B}$.

In [11], it was shown that X-CSP($\mathcal{B}$) exhibits trichotomy as $\mathcal{B}$ ranges over undirected, irreflexive cycles, with each problem being in either **L**, **NP**-complete or **PSPACE**-complete. The following classification was given for cliques.

Theorem 1. [11] *For $n \in \mathbb{N}$ and $X \subseteq \{1, \ldots, n\}$:*

(i) *X-CSP($\mathbb{K}_n$) is in* **L** *if $n \leq 2$ or $X \cap \big\{1, \ldots, \lfloor n/2 \rfloor\big\} = \emptyset$.*
(ii) *X-CSP($\mathbb{K}_n$) is* **NP***-complete if $n > 2$ and $X = \{1\}$.*
(iii) *X-CSP($\mathbb{K}_n$) is* **PSPACE***-complete if $n > 2$ and either $j \in X$ for $1 < j < n/2$ or $\{1, j\} \subseteq X$ for $j \in \big\{\lceil n/2 \rceil, \ldots, n\big\}$.*

Precisely the cases $\{j\}$-CSP($\mathbb{K}_{2j}$) are left open here. Of course, $\{1\}$-CSP($\mathbb{K}_2$) is *graph 2-colorability* and is in **L**, but for $j > 1$ the situation was very unclear, and the referees noted specifically this lacuna.

In this paper we **settle this question**, and find the surprising situation that $\{2\}$-CSP($\mathbb{K}_4$) is in **P** while $\{j\}$-CSP($\mathbb{K}_{2j}$) is **PSPACE**-complete for $j \geq 3$. The algorithm for the case $\{2\}$-CSP($\mathbb{K}_4$) is specialized and non-trivial, and consists in iteratively constructing a collection of forcing triples where we proceed to look for a contradiction.

As a **second focus** of the paper, we continue the study of $\{1, 2\}$-CSP(H). In particular, we focus on finite undirected graphs for which a dichotomy was proposed in [11]. As a fundamental step towards this, we first investigate the complexity of $\{1, 2\}$-CSP($\mathbb{P}_\infty$), where $\mathbb{P}_\infty$ denotes the infinite undirected path. We find tractability here in describing a particular unique obstruction, which takes the form of a special walk, whose presence or absence yields the answer to the problem. Again the algorithm is specialized and non-trivial, and in carefully augmenting it, we construct another polynomial-time algorithm, this time for all finite paths. This then proves the following theorem.

Theorem 2. *$\{1, 2\}$-CSP($\mathbb{P}_n$) is in* **P***, for each $n \in \mathbb{N}$.*

A corollary of this is the following **key result**.

Corollary 1. *$\{1,2\}$-CSP(H) is in* **P**, *for each forest H.*

Combined with the results from [8, 11], this allows us to observe a dichotomy for $\{1,2\}$-CSP(H) as H ranges over undirected graphs, each problem being either in **P** or **NP**-hard, in turn **settling a conjecture** proposed in [11].

Corollary 2. *Let H be a graph.*

(i) $\{1,2\}$-CSP(H) is in **P** *if H is a forest or is bipartite with a 4-cycle,*
(ii) $\{1,2\}$-CSP(H) is **NP***-hard in all other cases.*

In [11], the main preoccupation was in the distinction between **P** and **NP**-hard. Here we concentrate our observations to show situations in which we have sharp dichotomies between **P** and **PSPACE**-complete. In particular, for bipartite graphs, we are able to strengthen the above results in the following manner.

Theorem 3. *Let H be a bipartite graph.*

(i) $\{1,2\}$-CSP(H) is in **P** *if H is a forest or is bipartite with a 4-cycle,*
(ii) $\{1,2\}$-CSP(H) is **PSPACE***-complete in all other cases.*

Note that this cannot be strengthened further for non-bipartite graphs, since there are **NP**-complete cases (for instance when H is the octahedron $\mathbb{K}_{2,2,2}$) and the situation regarding the **NP**-complete cases is less clear.

Taken together, our work seems to indicate a rich and largely uncharted complexity landscape that these types of problems constitute. The associated combinatorics to this landscape appears quite complex and the absence of a simple algebraic approach is telling. We will return to the question of algebra in the final remarks of the paper.

The paper is structured as follows. In §2, we describe a characterization and a polynomial time algorithm for $\{2\}$-CSP$(\mathbb{K}_4)$. In §3, we show **PSPACE**-hardness for $\{n\}$-CSP$(\mathbb{K}_{2n})$ for $n \geq 3$. In §4, we characterize $\{1,2\}$-CSP for the infinite path $\mathbb{P}_\infty$ and describe the resulting polynomial algorithm. Then, in §5, we generalize this to finite paths and prove Theorem 2 and associated corollaries. Subsequently, in §6, we discuss the **P**/**PSPACE**-complete dichotomy of bipartite graphs, under $\{1,2\}$-CSP. Finally in §7, we illustrate some situations in which the intermediate **NP**-completeness arises by discussing cases with loops on vertices. We conclude the paper in §8 by giving some final thoughts.

1.1 Preliminaries

Our proofs use the game characterization and structural interpretation from [11]. For completeness, we summarize it here. This is as follows.

Given an input Ψ for X-CSP$(\mathcal{B})$, we define the following game $\mathscr{G}(\Psi, \mathcal{B})$:

Definition 1. *Let $\Psi := Q_1 x_1 Q_2 x_2 \dots Q_m x_m \; \psi(x_1, x_2, \dots, x_m)$. Working from the outside in, coming to a quantified variable $\exists^{\geq j} x$, the Prover (female) picks a subset B_x of j elements of B as witnesses for x, and an Adversary (male) chooses one of these, say b_x, to be the value of x, denoted by $f(x)$.*

Prover wins if f is a homomorphism to $\mathcal{B}$, i.e., if $\mathcal{B} \models \psi(f(x_1), f(x_2), \dots, f(x_m))$.

Lemma 1. *Prover has a winning strategy in the game $\mathscr{G}(\Psi, \mathcal{B})$ iff $\mathcal{B} \models \Psi$.*

Definition 2. *Let H be a graph. For an instance Ψ of X-CSP(H):*

- *define $\mathcal{D}_\psi$ to be the graph whose vertices are the variables of Ψ and edges are between variables v_i, v_j for which $E(v_i, v_j)$ appears in Ψ.*
- *denote $\prec$ the total order of variables of Ψ as they are quantified in the formula (from left to right).*

We follow the customary graph-theoretical notation with $V(G)$, $E(G)$ denoting the vertex set and edge set of a graph G, and $\mathbb{K}_n$, $\mathbb{C}_n$, and $\mathbb{P}_n$ denoting respectively the complete graph (clique), the cycle, and the path on n vertices.

2 Algorithm for $\{2\}$-CSP($\mathbb{K}_4$)

Theorem 4. *$\{2\}$-CSP($\mathbb{K}_4$) is decidable in polynomial time.*

The template $\mathbb{K}_4$ has vertices $\{1, 2, 3, 4\}$ and all possible edges between distinct vertices. Consider the instance Ψ of $\{2\}$-CSP($\mathbb{K}_4$) as a graph $G = \mathcal{D}_\psi$ together with a linear ordering $\prec$ on $V(G)$ (see Definition 2).

We iteratively construct the following three sets: R^+, R^-, and F. The set F will be a collection of unordered pairs of vertices of G, while R^+ and R^- will consist of unordered triples of vertices. (For simplicity we write $xy \in F$ in place of $\{x, y\} \in F$, and write $xyz \in R^+$ or R^- in place of $\{x, y, z\} \in R^+$ or R^-.)

The meaning of these sets is as follows. A pair $xy \in F$ where $x \prec y$ indicates that Prover in order to win must offer values so that the value $f(x)$ chosen by Adversary for x is different from the value $f(y)$ chosen for y. A triple $xyz \in R^+$ where $x \prec y \prec z$ indicates that if Adversary chose $f(x) \neq f(y)$, then Prover must offer one (or both) of $f(x), f(y)$ for z. A triple $xyz \in R^-$ where $x \prec y \prec z$ tells us that Prover must offer values different from $f(x), f(y)$ if $f(x) \neq f(y)$.

With this, we describe how to iteratively compute the three sets F, R^+, R^-. We start by initializing the sets as follows: $F = E(G)$ and $R^+ = R^- = \emptyset$. Then we perform the following rules as long as possible:

(X1) If there are $x, y, z \in V(G)$ such that $\{x, y\} \prec z$ where $xz, yz \in F$, then add xyz into R^-.

(X2) If there are vertices $x, y, w, z \in V(G)$ such that $\{x, y, w\} \prec z$ with $wz \in F$ and $xyz \in R^-$, then add xyw into R^+.

(X3) If there are $x, y, w, z \in V(G)$ such that $\{x, y, w\} \prec z$ with $wz \in F$ and $xyz \in R^+$, then if $\{x, y\} \prec w$, then add xyw into R^-
else add xw and yw into F.

(X4) If there are vertices $x, y, w, z \in V(G)$ such that $\{x, w\} \prec y \prec z$ with $xyz \in R^+$ and $wyz \in R^-$, then add xw into F, and add xwy into R^+.

(X5) If there are vertices $x, y, w, z \in V(G)$ such that $\{x, y, w\} \prec z$ where either $xyz, wyz \in R^+$, or $xyz, wyz \in R^-$, then add xyw into R^+.

(X6) If there are vertices $x, y, q, w, z \in V(G)$ such that $\{x, y, w\} \prec q \prec z$ where either $xyz, wqz \in R^+$, or $xyz, wqz \in R^-$, then add xyw and xyq into R^+.

(X7) If there are vertices $x, y, q, w, z \in V(G)$ such that $\{x, y, w\} \prec q \prec z$ where either $xyz \in R^+$ and $wqz \in R^-$, or $xyz \in R^-$ and $wqz \in R^+$, then add xyq into R^-, and if $\{x, y\} \prec w$, also add xyw into R^-,
else add xw and yw into F.

Theorem 5. *The following are equivalent:*

(i) $\mathbb{K}_4 \models \Psi$

(ii) *Prover has a winning strategy in* $\mathscr{G}(\Psi, \mathbb{K}_4)$.

(iii) *Prover can play so that in every instance of the game, the resulting mapping* $f : V(G) \to \{1, 2, 3, 4\}$ *satisfies the following properties:*

(S1) *For every* $xy \in F$*, we have:* $f(x) \neq f(y)$.

(S2) *For every* $xyz \in R^+$ *such that* $x \prec y \prec z$*:*
$$\text{if } f(x) \neq f(y), \text{ then } f(z) \in \{f(x), f(y)\}.$$

(S3) *For every* $xyz \in R^-$ *such that* $x \prec y \prec z$*:*
$$\text{if } f(x) \neq f(y), \text{ then } f(z) \notin \{f(x), f(y)\}.$$

(iv) *there is no triple* xyz *in* R^+ *such that* $x \prec y \prec z$ *and (see Fig. 1)*
- $xz \in F$ *or* $yz \in F$,
- *or* $xwz \in R^-$ *for some* $w \prec z$ *(possibly* $w = y$*)*,
- *or* $ywz \in R^-$ *for some* $y \prec w \prec z$.

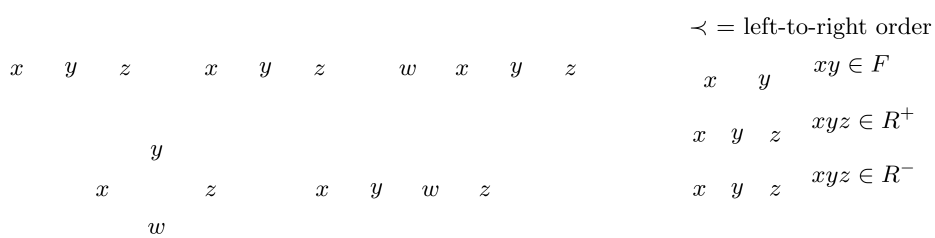

Fig. 1. Forbidden configurations from item (iv) of Theorem 5.

Proof. (Sketch) (i) $\iff$ **(ii)** is by definition. **(iii)**$\Rightarrow$**(ii)** is implied by the fact that $F \supseteq E(G)$, and that by (iii) Prover can play to satisfy (S1). Thus in every instance of the game the mapping f is a homomorphism of G to $\mathbb{K}_4$ $\Rightarrow$ (ii).

Then to complete the proof, we show the implications **(ii)⇒(iii)**, **(iii)⇒(iv)**, and **(iv)⇒(iii)**. This is done by analysis of possible cases.

For **(iii)⇒(iv)**, we show that in the presence of the obstruction from (iv), Adversary can play to violate (iii). For **(iv)⇒(iii)**, we let Prover make choices to satisfy (iii), first for triples in R^+, then triples in R^-, and finally edges in F. Assuming (iv), this will be a winning strategy. For **(ii)⇒(iii)**, we consider the vertex v where (iii) fails and choose v to be largest with respect to the order $\prec$. Assuming (ii) will imply an earlier such a vertex and lead to a contradiction. □

With this characterization, we can now prove Theorem 4 as follows.

Proof. (Theorem 4) By Theorem 5, it suffices to construct the sets F, R^+, and R^-, and check the conditions of item (iv) of the said theorem. This can clearly be accomplished in polynomial time, since each of the three sets contains at most n^3 elements, where n is the number of variables in the input formula, and elements are only added (never removed) from the sets. Thus either a new pair (triple) needs to be added as follows from one of the rules (X1)-(X7), or we can stop and the output the resulting sets. □

3 Hardness of $\{n\}$-CSP($\mathbb{K}_{2n}$) for $n \geq 3$

Theorem 6. *$\{n\}$-CSP($\mathbb{K}_{2n}$) is* **PSPACE**-*complete for all $n \geq 3$.*

The template $\mathbb{K}_{2n}$ consists of vertices $\{1, 2, \ldots, 2n\}$ and all possible edges between distinct vertices. We shall call these vertices *colours*. We describe a reduction from the **PSPACE**-complete [2] problem QCSP($\mathbb{K}_n$)=$\{1, n\}$-CSP($\mathbb{K}_n$) to $\{n\}$-CSP($\mathbb{K}_{2n}$). Consider an instance of QCSP($\mathbb{K}_n$), namely a formula Ψ where

$$\Psi = \exists^{\geq b_1} v_1 \, \exists^{\geq b_2} v_2 \, \ldots \, \exists^{\geq b_N} v_N \, \psi$$

where each $b_i \in \{1, n\}$. As usual (see Definition 2), let G denote the graph $\mathcal{D}_\psi$ with vertex set $\{v_1, \ldots, v_N\}$ and edge set $\{v_i v_j \mid E(v_i, v_j) \text{ appears in } \psi\}$.

We construct an instance Φ of $\{n\}$-CSP($\mathbb{K}_{2n}$) with the property that Ψ is a yes-instance of QCSP($\mathbb{K}_n$) if and only if Φ is a yes-instance of $\{n\}$-CSP($\mathbb{K}_{2n}$).

In short, we shall model the n-colouring using $2n-1$ colours, $n-1$ of which will treated as *don't care* colours (vertices coloured using any of such colours will be ignored). We make sure that the colourings where no vertex is assigned a don't-care colour precisely model all colourings that we need to check to verify that Ψ is a yes-instance.

We describe Φ by giving a graph H together with a total order of its vertices with the usual interpretation that the vertices are the variables of Φ, the total order is the order of quantification of the variables, and the edges of H define the conjunction of predicates $E(\cdot, \cdot)$ which forms the quantifier-free part ϕ of Φ.

We start constructing H by adding the vertices $v_1, v_2, \ldots, v_N$ and no edges. Then we add new vertices $u_1, u_2, \ldots, u_n$ and make them pairwise adjacent.

We make each v_i adjacent to u_1, and if $b_i = n$ (i.e. if v_i was quantified $\forall$), then we also make v_i adjacent to $u_2, u_3, \ldots, u_n$.

We complete H by introducing for each edge $xy \in E(G)$, a gadget consisting of new vertices w, q, z, a, b, c with edges wa, wb, qb, qc, za, zb, and we connect this gadget to the rest of the graph as follows: we make x adjacent to a, make y adjacent to b, make a adjacent to u_1, make c adjacent to u_1, u_2, u_3, and make each of a, b, c adjacent to $u_4, \ldots, u_n$. We refer to Figure 2 for an illustration.

The total order of $V(H)$ first lists $u_1, u_2, \ldots, u_n$, then $v_1, v_2, \ldots, v_N$ (exactly in the same order as quantified in Ψ), and then lists the remaining vertices of each gadget, in turn, as depicted in Figure 2 (listing w, q, z, a, b, c in this order).

We consider the game $\mathscr{G}(\Phi, \mathbb{K}_{2n})$ of Prover and Adversary played on Φ where Prover and Adversary take turns, for each variable in Φ in the order of quantification, respectively providing a set of n colours and choosing a colour from the set. Prover wins if this process leads to a proper $2n$-colouring of H (no adjacent vertices receive the same colour), otherwise Prover loses and Adversary wins. The formula Φ is a yes-instance if and only if Prover has a winning strategy.

Without loss of generality (up to renaming colours), we may assume that the vertices $u_1, u_2, \ldots, u_n$ get assigned colours $n+1, n+2, \ldots, 2n$, respectively, i.e. each u_i gets colour $n+i$. (The edges between these vertices make sure that Prover must offer distinct colours while Adversary has no way of forcing a conflict, since there are $2n$ colours available.)

The claim of Theorem 6 will then follow from the following two lemmas.

Lemma 2. *If Adversary is allowed to choose for the vertices x, y in the edge gadget (Figure 2) the same colour from $\{1, 2, \ldots, n\}$, then Adversary wins. If Adversary is allowed to choose $n+1$ for x or y, then Adversary also wins.*

In all other cases, Prover wins.

Lemma 3. *Φ is a yes-instance of $\{n\}$-CSP($\mathbb{K}_{2n}$) if and only if Ψ is a yes-instance of QCSP($\mathbb{K}_n$).*

We finish the proof by remarking that the construction of Φ is polynomial in the size of Ψ (in fact the reduction is in **L**). Thus, since QCSP($\mathbb{K}_n$) is **PSPACE**-hard, so is $\{n\}$-CSP($\mathbb{K}_{2n}$). This completes the proof of Theorem 6.

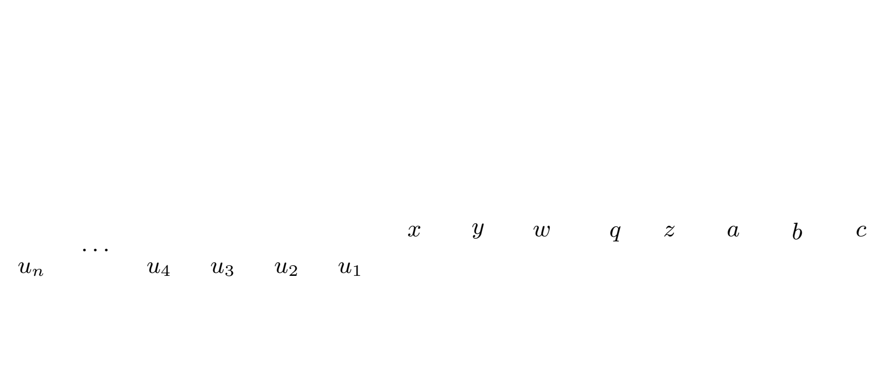

Fig. 2. The edge gadget (here, as an example, x is an $\exists$ vertex while y is a $\forall$ vertex).

4 Algorithm for $\{1,2\}$-CSP($\mathbb{P}_\infty$)

We consider the infinite path $\mathbb{P}_\infty$ to be the graph whose vertex set is $\mathbb{Z}$ and whose edges are $\{ij \;:\; |i-j|=1\}$. An instance to $\{1,2\}$-CSP($\mathbb{P}_\infty$) is a graph $G=\mathcal{D}_\psi$, a total order $\prec$ on $V(G)$, and a function $\beta : V(G) \to \{1,2\}$ where

$$\Psi := \exists^{\geq \beta(v_1)} v_1 \; \exists^{\geq \beta(v_2)} v_2 \; \cdots \; \exists^{\geq \beta(v_n)} v_n \bigwedge_{v_i v_j \in E(G)} E(v_i, v_j)$$

We write $X \prec Y$ if $x \prec y$ for each $x \in X$ and each $y \in Y$. Also, we write $x \prec Y$ in place of $\{x\} \prec Y$. A *walk* of G is a sequence $x_1, x_2, \ldots, x_r$ of vertices of G where $x_i x_{i+1} \in E(G)$ for all $i \in \{1, \ldots, r-1\}$. A walk $x_1, \ldots, x_r$ is a *closed walk* if $x_1 = x_r$. Write $|Q|$ to denote the *length* of the walk Q (number of edges on Q).

Definition 3. *If $Q = x_1, \ldots, x_r$ is a walk of G, we define $\lambda(Q)$ as follows:*

$$\lambda(Q) = |Q| - 2 \sum_{i=2}^{r-1} \big(\beta(x_i) - 1\big)$$

Put differently, we assign weights to the vertices of G, with weight $+1$ assigned to each $\exists^{\geq 2}$ node, and weight -1 to each $\exists^{\geq 1}$ node; the value $\lambda(Q)$ is then simply the total weight of all inner nodes in the walk Q.

Definition 4. *A walk $x_1, \ldots, x_r$ of G is a* looping walk *if $x_1 \neq x_r$ and if $r \geq 3$*

(i) $\{x_1, x_r\} \prec \{x_2, \ldots, x_{r-1}\}$, and
(ii) there is $\ell \notin \{1, r\}$ such that both $x_1, \ldots, x_\ell$ and $x_\ell, \ldots, x_r$ are looping walks.

The above is a recursive definition. Note that endpoints of a looping walk are distinct and never appear in the interior of the walk. Other vertices, however, may appear on the walk multiple times as long as the walk obeys (ii). Notably, it is possible that the same vertex is one of $x_2, \ldots, x_{\ell-1}$ as well as one of $x_{\ell-1}, \ldots, x_{r-1}$ where ℓ is as defined in (ii). See Figure 3 for examples.

Using looping walks, we define a notion of "distance" in G that will guide Prover in the game.

Definition 5. *For vertices $u, v \in V(G)$, define $\delta(u,v)$ to be the following:*
$\min\Big\{\lambda(Q) \;\Big|\; Q = x_1, \ldots, x_r \text{ is a looping walk of } G \text{ where } x_1 = u \text{ and } x_r = v\Big\}$.
If no looping walk between u and v exists, define $\delta(u,v) = \infty$.

In other words, $\delta(u,v)$ denotes the smallest λ-value of a looping walk between u and v. Note that $\delta(u,v) = \delta(v,u)$, since the definition of a looping walk does not prescribe the order of the endpoints of the walk.

The main structural obstruction in our characterization of is the following.

Definition 6. *A* bad walk *of G is a looping walk $Q = x_1, \ldots, x_r$ of G such that $x_1 \prec x_r$ and $\lambda(Q) \leq \beta(x_r) - 2$.*

4.1 Characterization

Theorem 7. *Suppose that G is a bipartite graph. Then the following statements are equivalent.*

(I) $\mathbb{P}_\infty \models \Psi$

(II) Prover has a winning strategy in $\mathscr{G}(\Psi, \mathbb{P}_\infty)$.

(III) Prover can play $\mathscr{G}(\Psi, \mathbb{P}_\infty)$ *so that in every instance of the game, the resulting mapping* f *satisfies the following for all* $u, v \in V(G)$ *with* $\delta(u,v) < \infty$:

$$|f(u) - f(v)| \leq \delta(u,v)\,, \tag{$\star$}$$

$$f(u) + f(v) + \delta(u,v) \text{ is an even number}. \tag{$\triangle$}$$

(IV) There are no $u, v \in V(G)$ *where* $u \prec v$ *such that* $\delta(u,v) \leq \beta(v) - 2$.

(V) There is no bad walk in G.

$\prec$ is the left-to-right order

$G:$

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
$\beta:$	$\exists$	$\exists^{\geq 2}$	$\exists^{\geq 2}$	$\exists^{\geq 2}$	$\exists$	$\exists^{\geq 2}$	$\exists$	$\exists$	$\exists^{\geq 2}$

Example looping walks:

$Q^* = v_1, v_9, v_8, v_7, v_2$ $\quad |Q^*| = 4 \quad \lambda(Q^*) = 4 - 2 \cdot 1 = 2$

$Q = v_1, v_9, v_8, v_7, v_6, v_5, v_4, v_3, v_4, v_5, v_6, v_7, v_2$ $\quad |Q| = 12$

$\{v_1, v_2\} \prec \{v_3, \ldots, v_9\}$ $\quad \lambda(Q) = 12 - 2 \cdot 6 = 0$

We decompose Q into looping walks:

$Q_1 = v_1, v_9, v_8, v_7, v_6, v_5, v_4, v_3$ $\quad \lambda(Q_1) = 7 - 2 \cdot 3 = 1$

$Q_2 = v_2, v_7, v_6, v_5, v_4, v_3$ $\quad \lambda(Q_2) = 5 - 2 \cdot 2 = 1$

$\{v_1, v_2\} \prec v_3 \prec \{v_4, \ldots, v_9\}$

Note that Q is a bad walk, while neither Q^* nor Q_1 nor Q_2 is.

Fig. 3. Examples of looping walks.

Proof. (Sketch) We prove the claim by considering individual implications. The equivalence (I)$\Leftrightarrow$(II) is proved as Lemma 1. The equivalence (IV)$\Leftrightarrow$(V) follows immediately from the definitions of $\delta(\cdot,\cdot)$ and bad walk. The other implications are proved as follows. For (III)$\Rightarrow$(II), we show that Prover's strategy described in (III) is a winning strategy. For (II)$\Rightarrow$(III), we show that every winning strategy must satisfy the conditions of (III). For (III)$\Rightarrow$(IV), we show that having vertices $u \prec v$ with $\delta(u,v) \leq \beta(v) - 2$ allows Adversary to win, by playing along the bad walk defined by vertices u, v. Finally, for (IV)$\Rightarrow$(III), assuming no bad pair u, v, we describe a Prover's strategy satisfying (III). $\square$

We conclude this section by remarking that the values $\delta(u,v)$ can be easily computed in polynomial time by dynamic programming. This allows us to test conditions of the above theorem and thus decide $\{1,2\}$-CSP$(\mathbb{P}_\infty)$ in polytime.

5 Algorithm for $\{1,2\}$-CSP($\mathbb{P}_n$)

The path $\mathbb{P}_n$ has vertices $\{1, 2, \ldots, n\}$ and edges $\{ij \ : \ |i-j| = 1\}$.

Let Ψ be an instance of $\{1,2\}$-CSP($\mathbb{P}_n$). As usual, let G be the graph $\mathcal{D}_\psi$ corresponding to Ψ, and let $\prec$ be the corresponding total ordering of $V(G)$.

For simplicity, let us assume that G is connected and bipartite with white and black vertices forming the bipartition. (If it is not bipartite, there is no solution; if disconnected, we solve the problem independently on each component.)

We start with a warmup lemma.

Lemma 4. *Assume $\mathbb{P}_\infty \models \Psi$. Let f be the first vertex in the ordering $\prec$. Then*

(i) $\mathbb{P}_1 \models \Psi \iff$ *G is the single $\exists^{\geq 1}$ vertex f.*
(ii) $\mathbb{P}_2 \models \Psi \iff$ *G does not contain $\exists^{\geq 2}$ vertex except possibly for f.*
(iii) $\mathbb{P}_3 \models \Psi \iff$ *all $\exists^{\geq 2}$ vertices in G have the same colour.*
(iv) $\mathbb{P}_4 \models \Psi \iff$ *all $\exists^{\geq 2}$ vertices in G are pairwise non-adjacent except possibly for f.*
(v) $\mathbb{P}_5 \models \Psi \iff$ *there is colour C (black or white) such that each edge xy between two $\exists^{\geq 2}$ vertices where $x \prec y$ is such that x has colour C.*

We now expand this lemma to the general case of $\{1,2\}$-CSP($\mathbb{P}_n$) as follows. Recall that we proved that $\mathbb{P}_\infty \models \Psi$ if and only if Prover can play $\mathscr{G}(\Psi, \mathbb{P}_\infty)$ so that in every instance of the game, the resulting mapping f satisfies ($\star$) and ($\triangle$). In fact the proof of (III)$\Rightarrow$(II) from Theorem 7 shows that every winning strategy of Prover has this property. We use this fact in the subsequent text.

The following value $\gamma(v)$ will allow us to keep track of the distance of $f(v)$ from the center of the (finite) path.

Definition 7. *For each vertex v we define $\gamma(v)$ recursively as follows:*

$$\gamma(v) = 0 \qquad \text{if } v \text{ is first in the ordering } \prec$$

$$\text{else} \quad \gamma(v) = \beta(v) - 1 + \max\left\{0, \max_{u \prec v}\Big(\gamma(u) - \delta(u,v) + \beta(v) - 1\Big)\right\}$$

Lemma 5. *Let M be a real number. Suppose that $\mathbb{P}_\infty \models \Psi$ and that Prover plays a winning strategy in the game $\mathscr{G}(\Psi, \mathbb{P}_\infty)$. Then Adversary can play so that the resulting mapping f satisfies $|f(v) - M| \geq \gamma(v)$ for every vertex $v \in V(D_\psi)$.*

Lemma 6. *Let M be a real number. Suppose that $\mathbb{P}_\infty \models \Psi$. Then there exists a winning strategy for Prover such that in every instance of the game the resulting mapping f satisfies $|f(v) - M| \leq \gamma(v) + 1$ for every $v \in V(\mathcal{D}_\psi)$.*

With these tools, we can now prove a characterization of the case of even n.

Theorem 8. *Let $n \geq 4$ be even. Assume that $\mathbb{P}_\infty \models \Psi$. Then TFAE.*

(I) $\mathbb{P}_n \models \Psi$.
(II) Prover has a winning strategy in the game $\mathscr{G}(\Psi, \mathbb{P}_n)$.
(III) There is no vertex v with $\gamma(v) \geq \frac{n}{2}$.

Proof. Note first that since n is even, we may assume, without loss of generality, the first vertex in the ordering is quantified $\exists^{\geq 1}$. If not, we can freely change its quantifier to $\exists^{\geq 1}$ without affecting the satisfiability of the intance.

(I)$\Leftrightarrow$(II) is by Lemma 1. For (II)$\Rightarrow$(III), assume there is v with $\gamma(v) \geq \frac{n}{2}$ and Prover has a winning strategy in $\mathscr{G}(\Psi, \mathbb{P}_n)$. This is also a winning strategy in $\mathscr{G}(\Psi, \mathbb{P}_\infty)$. This allows us to apply Lemma 5 for $M = \frac{n+1}{2}$ to conclude that Adversary can play against Prover so that $|f(v) - \frac{n+1}{2}| = |f(v) - M| \geq \gamma(v) \geq \frac{n}{2}$. Thus either $f(v) \geq \frac{2n+1}{2} > n$ or $f(v) \leq \frac{1}{2} < 1$. But then $f(v) \notin \{1, \ldots, n\}$ contradicting our assumption that Prover plays a winning strategy.

For (III)$\Rightarrow$(II), assume that $\gamma(v) \leq \frac{n}{2} - 1$ for all vertices v. We apply Lemma 6 for $M = \frac{n+1}{2}$. This tells us that Prover has a winning strategy on $\mathscr{G}(\Psi, \mathbb{P}_\infty)$ such that in every instance of the game, if f is the resulting mapping, the mapping satisfies $|f(v) - \frac{n+1}{2}| \leq \gamma(v) + 1$ for every vertex v. From this we conclude that $f(v) \geq \frac{n+1}{2} - \gamma(v) - 1 \geq \frac{n+1}{2} - \frac{n}{2} = \frac{1}{2}$ and that $f(v) \leq \frac{2n+1}{2} = n + \frac{1}{2}$. Therefore $f(v) \in \{1, 2, \ldots, n\}$ confirming that f is a valid homomorphism to $\mathbb{P}_n$. $\square$

This generalizes to odd n with a subtle twist. Define $\gamma'(v)$ using same recursion as $\gamma(v)$ except set $\gamma'(v) = \beta(v) - 1$ if v is first in $\prec$. Note that $\gamma'(v) \geq \gamma(v)$.

Theorem 9. *Let $n \geq 5$ be odd. Assume that $\mathbb{P}_\infty \models \Psi$ and that the vertices of $\mathcal{D}_\psi$ are properly coloured with colours black and white. Then TFAE.*

- *(I) $\mathbb{P}_n \models \Psi$.*
- *(II) Prover has a winning strategy in the game $\mathscr{G}(\Psi, \mathbb{P}_n)$.*
- *(III) There are no vertices u, v with $\gamma'(u) \geq \frac{n-1}{2}$ and $\gamma'(v) \geq \frac{n-1}{2}$ such that u is black and v is white.*

Now to derive Theorem 2, it remains to observe that the values $\gamma(v)$ and $\gamma'(v)$ can be calculated using dynamic programming in polynomial time.

5.1 Proofs of Corollaries 1 and 2

In this section, we sketch proofs of the two corollaries.

For Corollary 1, we want to decide $\{1, 2\}$-CSP(H) when H is a forest. Let Ψ be a given instance to this problem, and let $G = \mathcal{D}_\psi$ be the corresponding graph.

First, we note that we may assume that H is a tree. This follows easily (with a small caveat mentioned below) as the connected components of G have to be mapped to connected components of H. Therefore with H being a tree, we first claim that if Ψ is a yes-instance, then Ψ is also a yes-instance to $\{1, 2\}$-CSP($\mathbb{P}_\infty$). To conclude this, it can be shown that the condition (III) of Theorem 7 can be generalized to trees by replacing the absolute value in the condition ($\star$) by the distance in H, and by using a proper colouring of H instead of parity in ($\triangle$). This implies that no two vertices u,v are mapped in H farther away than $\delta(u, v)$. So a bad walk cannot exist and Ψ is a yes-instance of $\{1, 2\}$-CSP($\mathbb{P}_\infty$).

A similar argument allows us to generalize Theorems 8 and 9 to trees. Namely, in an optimal strategy Adversary will play **away** from some vertex, while Prover

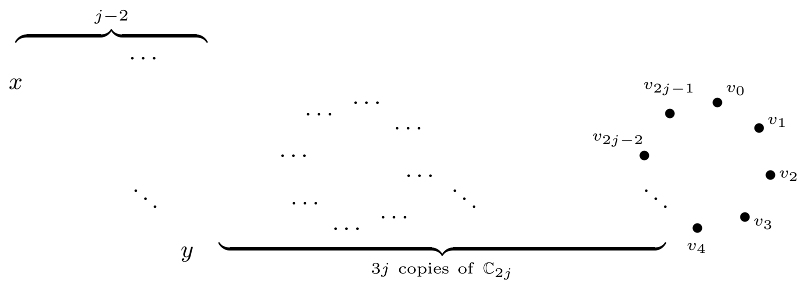

Fig. 4. The gadget for the case when H contains a cycle $\mathbb{C}_{2j}$.

will play **towards** some vertex. The absolute values will again be replaced by distances in H. From this we conclude that Adversary can force each v to be assigned to a vertex in H which is at least $\gamma'(v)$ or $\gamma(v)$ away from the center vertex, resp. center edge of H. In summary, this then proves the following.

Corollary 3. *Let H be a tree. Let P be a longest path in H. Then Ψ is a yes-instance of $\{1,2\}$-CSP(H) if and only if Ψ is a yes-instance of $\{1,2\}$-CSP(P).*

This can be phrased more generally for forests in a straightforward manner. The only caveat is that if two components contain a longest path with odd number of vertices, then we can make the first vertex in the instance an $\exists^{\geq 1}$ vertex without affecting the satisfiability, because if it is $\exists^{\geq 2}$, we let Adversary choose which midpoint of the two longest paths to use (and either choice is fine).

Finally, to prove Corollary 2, we note that $\{1,2\}$-CSP(H) is **NP**-hard for non-bipartite H, since $\{1\}$-CSP(H) is as famously proved in [8]. For bipartite H, the problem is in **P** if H is a forest (Corollary 1) or if H contains a 4-cycle (Proposition 10 in [11]). For bipartite graphs of larger girth, the problem is actually **PSPACE**-complete as we prove in the next section (Proposition 1).

6 Proof of Theorem 3

In this section, we prove the **P** / **PSPACE** dichotomy for $\{1,2\}$-CSP(H) for bipartite graphs H as stated in Theorem 3. We have already discussed the polynomial cases in the previous section. It remains to discuss the hardness.

Proposition 1. *If H is a bipartite graph whose smallest cycle is $\mathbb{C}_{2j}$ for $j \geq 3$, then $\{1,2\}$-CSP(H) is* **PSPACE**-*complete.*

Proof. We reuse the reduction from [11] used to prove Theorem 1. We briefly discuss the key steps. The reduction is from QCSP($\mathbb{K}_j$). Let Ψ be an input formula for QCSP($\mathbb{K}_j$). We begin by considering the graph $\mathcal{D}_\psi$ to which we add a disjoint copy $W = \{w_1, \ldots, w_{2j}\}$ of $\mathbb{C}_{2j}$. Then we replace every edge $(x,y) \in \mathcal{D}_\psi$ with a gadget shown in Figure 4, where the black vertices are identified with W. Finally, for $\forall$ variables v of Ψ, we add a new path $z_1, z_2, \ldots, z_j$ where $z_j = v$.

The resulting graph defines the quantifier-free part of θ of our desired formula Θ. The quantification in Θ is as follows. The outermost quantifiers are $\exists^{\geq 2}$ for variables $w_1, \ldots, w_{2j}$. Then we move inwards through the quantifier order of Ψ; when we encounter an existential variable v, we apply $\exists^{\geq 1}$ to it in Θ. When we encounter a $\forall$ variable v, we apply $\exists^{\geq 2}$ to the path $z_1, z_2, \ldots, z_j$ constructed for v, in that order. All the remaining variables are then quantified $\exists^{\geq 1}$.

As proved in [11], the cycle $\mathbb{C}_{2j}$ models Θ if and only if $\mathbb{K}_j$ models Ψ. We now adjust this to the bipartite graph H. There are three difficulties arising from simply using the above construction as it is.

Firstly, assume the variables $w_1, \ldots, w_{2j}$ are mapped to a fixed copy C of $\mathbb{C}_{2j}$ in H. We need to ensure that variables x, y derived from the original instance Ψ are also mapped to C. For y variables in our gadget one can check this must be true – the successive cycles in the edge gadget may never deviate from C, since H contains no 4-cycle. For x variables off on the pendant this might not be true. To fix this, we insist that Ψ contains an atom $E(x,y)$ iff it also contains $E(y,x)$; QCSP($\mathbb{K}_j$) remains **PSPACE**-complete on such instances [2].

Secondly, we need to check that Adversary has freedom to assign any value from C to each $\forall$ variable v. Consider $z_1, \ldots, z_j$, the path associated with v. As long as Prover offers values for $z_1, \ldots, z_j$ from C, Adversary has freedom to chose any value for $v = z_j$. If on the other hand Prover offers for one of $z_1, \ldots, z_j$, say for z_i, a value not on C, then Adversary can choose all subsequent $z_{i+1}, \ldots, z_j$ to also be mapped outside C, since H has no cycle shorter than $\mathbb{C}_{2j}$. Thus $v = z_j$ is mapped outside C, but we already ensured that this does not happen.

Finally, we discuss how to ensure that W is mapped to a copy of $\mathbb{C}_{2j}$. Since each vertex in W is quantified $\exists^{\geq 2}$, Adversary can force this by always choosing a value not seen already when going through each of $w_1, \ldots, w_{2j}$ in turn. If this is not possible (both offered values have been seen), this gives rise to a cycle in H shorter than $\mathbb{C}_{2j}$. In conclusion, if Adversary maps W to a cycle, then Prover must play exclusively on this cycle, thus solving QCSP($\mathbb{K}_j$). If Adversary maps W to a subpath of $\mathbb{C}_{2j}$, then Prover can play to win (regardless whether Φ is a yes- or no- instance). So the situation is just like with $\{1,2\}$-CSP($\mathbb{C}_{2j}$). □

7 Partially reflexive graphs

In this section, we briefly list some results for graphs allowing self-loops on some vertices (so-called *partially reflexive* graphs). Our understanding of these cases is rather limited and some recent results [10, 12] suggest that a simple dichotomy is very unlikely. Nonetheless, some cases might still be of further interest.

First, we consider the class of undirected graphs with a single dominating vertex w which is also a self-loop.

Proposition 2. *If H has a reflexive dominating vertex w and $H \setminus \{w\}$ contains a loop or is irreflexive bipartite, then $\{1,2\}$-CSP(H) is in* **P**.

Proposition 3. *If H has a reflexive dominating vertex w and $H \setminus \{w\}$ is irreflexive non-bipartite, then $\{1,2\}$-CSP(H) is* **NP**-*complete.*

Corollary 4. *If H has a reflexive dominating vertex, then $\{1,2\}$-CSP(H) is either in* **P** *or is* **NP**-*complete.*

It follows from Proposition 3 that there is a partially reflexive graph on four vertices, $\mathbb{K}_4$ with a single reflexive vertex, so that the corresponding $\{1,2\}$-CSP is **NP**-complete. We can argue this phenomenem is not visible on smaller graphs.

Proposition 4. *Let H be a (partially reflexive) graph on at most three vertices, then either $\{1,2\}$-CSP(H) is in* **P** *or it is* **PSPACE***-complete.*

8 Final remarks

In this paper we have settled the major questions left open in [11] and it might reasonably be said we have now concluded our preliminary investigations into constraint satisfaction with counting quantifiers. Of course there is still a wide vista of work remaining, not the least of which is to improve our **P**/ **NP**-hard dichotomy for $\{1,2\}$-CSP on undirected graphs to a **P**/ **NP**-complete / **PSPACE**-complete trichotomy (if indeed the latter exists). The absence of a similar trichotomy for QCSP, together with our reliance on [8], suggests this could be a challenging task. Some more approachable questions include lower bounds for $\{2\}$-CSP$(\mathbb{K}_4)$ and $\{1,2\}$-CSP$(\mathbb{P}_\infty)$. For example, intutition suggests these might be **NL**-hard (even **P**-hard for the former). Another question would be to study X-CSP$(\mathbb{P}_\infty)$, for $\{1,2\} \nsubseteq X \subset \mathbb{N}$.

Since we initiated our work on constraint satisfaction with counting quantifiers, a possible algebraic approach has been published in [5, 6]. It is clear reading our expositions that the combinatorics associated with our counting quantifiers is complex, and unfortunately the same seems to be the case on the algebraic side (where the relevant "expanding" polymorphisms have not previously been studied in their own right). At present, no simple algebraic method, generalizing results from [2], is known for counting quantifiers with majority operations. This would be significant as it might help simplify our tractability result of Theorem 2. So far, only the Mal'tsev case shows promise in this direction.

References

1. Barto, L., Kozik, M., and Niven, T. The CSP dichotomy holds for digraphs with no sources and no sinks (a positive answer to a conjecture of Bang-Jensen and Hell). *SIAM Journal on Computing 38*, 5 (2009), 1782–1802.
2. Börner, F., Bulatov, A. A., Chen, H., Jeavons, P., and Krokhin, A. A. The complexity of constraint satisfaction games and QCSP. *Inf. Comput. 207*, 9 (2009), 923–944.
3. Bulatov, A. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *J. ACM 53*, 1 (2006), 66–120.
4. Bulatov, A., Krokhin, A., and Jeavons, P. G. Classifying the complexity of constraints using finite algebras. *SIAM Journal on Computing 34* (2005), 720–742.
5. Bulatov, A. A., and Hedayaty, A. Counting predicates, subset surjective functions, and counting csps. In *42nd IEEE International Symposium on Multiple-Valued Logic, ISMVL 2012* (2012), pp. 331–336.
6. Bulatov, A. A., and Hedayaty, A. Galois correspondence for counting quantifiers. *CoRR abs/1210.3344* (2012).

7. Feder, T., and Vardi, M. The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory. *SIAM Journal on Computing 28* (1999), 57–104.
8. Hell, P., and Nešetřil, J. On the complexity of H-coloring. *Journal of Combinatorial Theory, Series B 48* (1990), 92–110.
9. Kolaitis, P. G., and Vardi, M. Y. *Finite Model Theory and Its Applications (Texts in Theoretical Computer Science. An EATCS Series)*. Springer-Verlag New York, Inc., 2005, ch. A logical Approach to Constraint Satisfaction.
10. Madelaine, F. R., and Martin, B. QCSP on partially reflexive cycles - the wavy line of tractability. In *8th International Computer Science Symposium in Russia, CSR 2013* (2013), pp. 322–333.
11. Madelaine, F. R., Martin, B., and Stacho, J. Constraint satisfaction with counting quantifiers. In *7th International Computer Science Symposium in Russia, CSR 2012* (2012), pp. 253–265.
12. Martin, B. QCSP on partially reflexive forests. In *Principles and Practice of Constraint Programming - 17th International Conference, CP 2011* (2011).
13. Schaefer, T. J. The complexity of satisfiability problems. In *Proceedings of STOC'78* (1978), pp. 216–226.