AN EFFECTIVE DICHOTOMY FOR THE COUNTING CONSTRAINT SATISFACTION PROBLEM

MARTIN DYER[*] AND DAVID RICHERBY[†]

Abstract. Bulatov (2008) gave a dichotomy for the counting constraint satisfaction problem #CSP. A problem from #CSP is characterised by a constraint language Γ, a fixed, finite set of relations over a finite domain D. An instance of the problem uses these relations to constrain an arbitrarily large finite set of variables. Bulatov showed that the problem of counting the satisfying assignments of instances of any problem from #CSP is either in polynomial time (FP) or is #P-complete. His proof draws heavily on techniques from universal algebra and cannot be understood without a secure grasp of that field. We give an elementary proof of Bulatov's dichotomy, based on succinct representations, which we call *frames*, of a class of highly structured relations, which we call *strongly rectangular.* We show that these are precisely the relations which are invariant under a *Mal'tsev polymorphism.* En route, we give a simplification of a decision algorithm for strongly rectangular constraint languages, due to Bulatov and Dalmau (2006). We establish a new criterion for the #CSP dichotomy, which we call *strong balance*, and we prove that this property is decidable. In fact, we establish membership in NP. Thus, we show that the dichotomy is effective, resolving the most important open question concerning the #CSP dichotomy.

1. Introduction. The constraint satisfaction problem (CSP) is ubiquitous in computer science. Problems in such diverse areas as Boolean logic, graph theory, database query evaluation, type inference, scheduling and artificial intelligence can be expressed naturally in the setting of assigning values from some domain to a collection of variables, subject to constraints on the combinations of values taken by given tuples of variables [17]. CSP is directly equivalent to the problem of evaluating conjunctive queries on databases [22] and to the homomorphism problem for relational structures [17]. Weighted versions of CSP appear in statistical physics, where the total weight of solutions corresponds to the so-called partition function of a spin system [16].

For example, suppose we wish to know if a graph is 3-colourable. The question we are trying to answer is whether we can assign a colour (domain value) to each vertex (variable) such that, whenever two vertices are adjacent in the graph, they receive a different colour (constraints). Similarly, by asking if a 3-CNF formula is satisfiable, we are asking if we can assign a truth value to each variable such that every clause contains at least one true literal.

Since it includes both 3-COLOURABILITY and 3-SAT, this general form of the CSP, known as *uniform* CSP, is NP-complete. Therefore, attention has focused on *nonuniform* CSP. Here, we fix a domain and a finite *constraint language* Γ, a set of relations over that domain. Having fixed Γ, we only allow constraints of the form, "the values assigned to the variables $v_1, \dots, v_r$ must be a tuple in the r-ary relation $R \in \Gamma$" (we define these terms formally in Section 2). We write CSP(Γ) to denote nonuniform CSP with constraint language Γ. To express 3-COLOURABILITY in this setting, we just take Γ to be the disequality relation on a set of three colours. 3-SAT is also expressible: to see this, observe that, for example, the clause $\neg x \vee y \vee \neg z$ corresponds to the relation $\{\mathtt{t},\mathtt{f}\}^3 \setminus \{\mathtt{t},\mathtt{f},\mathtt{t}\}$, where t indicates "true" and f "false", and that the other seven patterns of negations within a clause can be expressed similarly.

Thus, there are languages Γ for which CSP(Γ) is NP-complete. Of course, we can also express polynomial-time problems such as 2-COLOURABILITY and 2-SAT. Feder

[*] School of Computing, University of Leeds, Leeds, LS2 9JT, UK.

[†] Department of Computer Science, University of Liverpool, Liverpool, L69 3BX, UK.

This research was supported by EPSRC grants EP/E062172/1 "The Complexity of Counting in Constraint Satisfaction Problems" and EP/I012087/1 "Computational Counting".

and Vardi [17] conjectured that these are the only possibilities: that is, for all Γ, $\mathsf{CSP}(\Gamma)$ is in P or is NP-complete. To date, this conjecture remains open but it is known to hold in special cases [1, 20, 26]. Recent efforts to resolve the conjecture have focused on techniques from universal algebra [12].

There can be no dichotomy for the whole of NP, since Ladner [23] has shown that either $\mathsf{P} = \mathsf{NP}$ or there is an infinite hierarchy of complexity classes between them. Hence, assuming that $\mathsf{P} \neq \mathsf{NP}$, there exist problems in NP that are neither complete for the class nor in P. However, it is not unreasonable to conjecture a dichotomy for CSP, since there are NP problems, such as graph Hamiltonicity and even connectivity, that cannot be expressed as $\mathsf{CSP}(\Gamma)$ for any finite Γ. This follows from the observation that any set S of structures (e.g., graphs) that is definable in CSP has the property that, if $A \in S$ and there is a homomorphism $B \to A$, then $B \in S$; neither the set of Hamiltonian nor connected graphs has this property. Further, Ladner's theorem is proven by a diagonalisation that does not seem to be expressible in CSP [17].

In this paper, we consider the *counting* version of $\mathsf{CSP}(\Gamma)$, which we denote $\#\mathsf{CSP}(\Gamma)$. Rather than ask whether an instance of $\mathsf{CSP}(\Gamma)$ has a satisfying assignment, we ask how many satisfying assignments there are. The corresponding conjecture was that, for every Γ, $\#\mathsf{CSP}(\Gamma)$ is either computable in polynomial time or complete for $\#\mathsf{P}$. We give formal definitions in the next section but, informally, $\#\mathsf{P}$ is the analogue of NP for counting problems. Again, a modification of Ladner's proof shows that there can be no dichotomy for the whole of $\#\mathsf{P}$. Note that the decision version of any problem in NP is trivially reducible to the corresponding counting problem in $\#\mathsf{P}$: if we can count the number of solutions, we can certainly determine whether one exists. However, the converse cannot hold under standard assumptions about complexity theory: there are well-known polynomial-time algorithms that determine whether a graph admits a perfect matching but it is $\#\mathsf{P}$-complete to count the perfect matchings of even a bipartite graph [28].

Dichotomies for $\#\mathsf{CSP}(\Gamma)$ are known in several special cases [10, 11, 13, 15, 16], each consistent with the conjecture that $\#\mathsf{CSP}(\Gamma)$ is always either polynomial-time computable or $\#\mathsf{P}$-complete. However, Bulatov recently made a major breakthrough by proving a dichotomy for all Γ [2, 3].

Bulatov's proof makes heavy use of the techniques of universal algebra. A relation is said to be *pp-definable* over a constraint language Γ if it can be defined from the relations in Γ by a logical formula that uses only conjunction and existential quantification. Geiger [19] showed that an algebra can be associated with the set of pp-definable relations over Γ and Bulatov examines detailed properties of the *congruence lattice* of this algebra.[1] The structure of quotients in this lattice must have certain algebraic properties, which can be derived from *tame congruence theory* [21] and *commutator theory* [18]. Bulatov constructs an algorithm for the polynomial-time cases, based on decomposing this congruence lattice and using the structure of its quotients. However, he is only able to do this, in general, by transforming the relation corresponding to the input instance to one which is a *subdirect power.* It is even nontrivial to prove that this transformation inherits the required property of the original. His paper runs to some 43 pages and is very difficult to follow for anyone who is not expert in these areas. The criterion of Bulatov's dichotomy, which is based on infinite algebras constructed from Γ, was not shown to be decidable. It also seems difficult to apply his criterion to recover the special cases mentioned above.

Our main results are a new and elementary proof of Bulatov's theorem and a

[1] We will not define these terms from universal algebra, as they are not needed for our analysis.

proof that the dichotomy is effective. Thus, we answer, in the affirmative, the major open question in [3]. We follow Bulatov's approach by working with the relation over Γ determined by the input, but we require almost no machinery from universal algebra. The little that is used is defined and explained below. We develop a different criterion for the #CSP dichotomy, *strong balance*, which is based on properties of ternary relations definable in the constraint language. We show that it is equivalent to Bulatov's *congruence singularity* criterion.

Using strong balance, we construct a relatively simple iterative algorithm for the polynomial-time cases, which requires no algebraic properties. In fact, the bound on the time complexity of our counting algorithm is no worse than that for deciding if the input has satisfying assignments.

We then use our criterion to prove decidability of the #CSP dichotomy. We show that deciding strong balance is in NP, where the input size is that of Γ. Of course, complexity is not a central issue in the nonuniform model of #CSP, since Γ is considered to be a constant. It is only decidability that is important. However, the complexity of deciding the dichotomy seems an interesting computational problem in its own right.

1.1. Our proofs. Our proofs are almost entirely self-contained and should be accessible to readers with no knowledge of universal algebra and very little background in CSP. We use reductions from two previous papers on counting complexity, by Dyer and Greenhill [16] and by Bulatov and Grohe [8]. We also use results from Bulatov and Dalmau [6], but we include short proofs of these. The papers [6, 8] deal partly with ideas from universal algebra, but we make no use of those ideas. We use only one idea from universal algebra, that of a *Mal'tsev polymorphism.* This will be defined and explained in Section 2 below.

The proof is based around a succinct representation for relations preserved by a Mal'tsev polymorphism. We call such relations *strongly rectangular* for reasons which will become clear. Our representation is called a *frame*, and is similar to the *compact representation* of Bulatov and Dalmau [4]. Frames are smaller than compact representations, since they avoid some redundancy in the representation.

We define a *frame* for a relation $R \subseteq D^n$ to be a relation $F \subseteq R$ with the following two properties. First, whenever R contains a tuple with ith component a, F also contains such a tuple. Second, for $1 < i \leq n$ say that a set $S \subseteq D$ is i-equivalent in R if R contains tuples which agree on their first $i - 1$ elements and whose ith elements are exactly the members of S. Any set that is i-equivalent in R must also be i-equivalent in F, but note that there may be several common prefixes for S in R when only one is required in F. We show that every n-ary strongly rectangular relation over D has a *small* frame of cardinality at most $|D|n$, whereas R may have cardinality up to $|D|^n$. Further, we show how to construct such a frame efficiently and how to recover a strongly rectangular relation R from any of its frames.

Now, suppose we have an instance Φ of #CSP(Γ) for some strongly rectangular constraint language Γ, with m constraints in n variables. Using methods similar to those of Bulatov and Dalmau [4], we construct a frame for the solution set of Φ in polynomial time, by starting with a frame for D^n and introducing the constraints one at a time. A frame is empty if, and only if, it represents the empty relation so, at this point, we have re-proven Bulatov and Dalmau's result that there is a polynomial-time algorithm for the decision problem CSP(Γ) for any strongly rectangular constraint language Γ. We give an explicit time complexity for this algorithm, which is $\mathcal{O}(mn^4)$ for fixed Γ. Bulatov and Dalmau [4] gave no time estimate, showing only that their procedure runs in polynomial time.

Any ternary relation $R \subseteq A_1 \times A_2 \times A_3$ (where the A_i need not be disjoint) induces a matrix $M = (m_{xy})$ with rows and columns indexed by A_1 and A_2 and with

$$m_{xy} = |\{z : (x, y, z) \in R\}|\,.$$

We say that R is *balanced* if M's rows and columns can be permuted to give a block-diagonal matrix in which every block has rank one, and that a relation $R \subseteq D^n$ for any $n > 3$ is balanced if every expression of it as a ternary relation in $D^k \times D^\ell \times D^m$ ($k, \ell, m \geq 1$, $k + \ell + m = n$) is balanced. A constraint language Γ is *strongly balanced* if every relation of arity three or more that is pp-definable relation over Γ is balanced. Via a brief detour through weighted #CSP, we show that #$\mathsf{CSP}(\Gamma)$ is #P-complete if Γ is not strongly balanced.

If Γ is strongly balanced, we compute the number of satisfying assignments to a $\mathsf{CSP}(\Gamma)$ instance as follows. Let $R \subseteq D^n$ be the set of satisfying assignments. First, we construct a small frame F for R, as above. If R is unary, we have $F = R$ so we can trivially compute $|R|$.

Otherwise, for $1 \leq i < j \leq n$, let $N_{i,j}(a)$ be the number of prefixes $u_1 \dots u_i$ such that there is a tuple $u_1 \dots u_n \in R$ with $u_j = a$. In particular, then, summing the values of $N_{n-1,n}(-)$ gives $|R|$. Since the functions $N_{1,j}$ can be calculated easily from the frame, we just need to show how to compute $N_{i,j}$ for each $j > i$, given $N_{i-1,j}$ for each $j \geq i$. Writing $[k]$ for the set $\{1, \dots, k\}$, we can consider the set $\mathsf{pr}_{[i] \cup \{j\}} R$ to be a ternary relation on $\mathsf{pr}_{[i-1]} R \times \mathsf{pr}_i R \times \mathsf{pr}_j R$. R is strongly balanced so the matrix given by $M_{xy} = |\{\mathbf{u} : (\mathbf{u}, x, y) \in \mathsf{pr}_{[i] \cup \{j\}} R\}|$ is a rank-one block matrix and the sum of the a-indexed column of the matrix is $N_{i,j}(a)$.

By taking quotients with respect to certain congruences, we obtain another rank-one block matrix $\widehat{M}$, whose block structure and row and column sums we can determine. A key fact about rank-one block matrices is that this information is sufficient to recover the entries of the matrix. This allows us to recover M and, hence, compute the values $N_{i,j}(a)$ for each j and a. Iterating, we can determine the function $N_{n-1,n}$ and, hence, compute $|R|$.

Finally, we show that the strong balance property is decidable. Our proof of decidability rests on showing that, if Γ is not strongly balanced, then there is a counterexample with a number of variables that is only polynomial in the size of Γ. We do this by reformulating the strong balance criterion for a given formula Ψ as a question concerning counting assignments in a formula derived from Ψ. This reformulation enables us to apply a technique of Lovász [24]. The technique further allows us to recast strong balance in terms of the symmetries of a fixed structure, that is easily computable from Γ. We are thus able to show that deciding strong balance is in NP, where the input size is that of Γ.

1.2. Organisation of the paper. The remainder of the paper is organised as follows. Preliminary definitions and notation are given in Section 2. In Section 3, we define the notion of strong rectangularity that we use throughout the paper and, in Section 4, we further study the properties of strongly rectangular relations and introduce frames, our succinct representations of such relations. We give an efficient procedure for constructing frames in Section 5. In Section 6, we introduce counting problems and, in Section 7, we define the key notion of a strongly balanced constraint language and prove that #$\mathsf{CSP}(\Gamma)$ is solvable in polynomial time if Γ is strongly balanced and is #P-complete otherwise. In Section 8, we show that our dichotomy is decidable, in fact in the complexity class NP. Some concluding remarks appear in Section 9.

2. Definitions and notation. In this section, we present the definitions and notation used throughout the paper. We defer to Section 8 material relating to certain classes of functions that are used only in that section.

For any natural number n, we write $[n]$ for the set $\{1,\dots,n\}$.

2.1. Relations and constraints. Let $D = \{d_1, d_2, \dots, d_q\}$ be a finite *domain* with $q = |D|$. We will always consider q to be a constant and we assume that $q \geq 2$ to avoid trivialities. A *constraint language* Γ is a finite set of finitary relations on D, including the binary equality relation $\{(d_i, d_i) : i \in [q]\}$, which we denote by $=$. We will call $\mathfrak{G} = (D, \Gamma)$ a *relational structure*. We may view an r-ary relation H on D with $\ell = |H|$ as an $\ell \times r$ matrix with elements in D. Then a tuple $\mathbf{t} \in H$ is any row of this matrix. We will usually write tuples in the standard notation, for example $(t_1, t_2, \dots, t_r)$. For brevity, however, we also write tuples in string notation, for example, $t_1 t_2 \dots t_r$, where this can cause no confusion.

If R is an n-ary relation and $\mathbf{i} = (i_1, \dots, i_k)$ are distinct elements of $[n]$, we write $\mathsf{pr}_{\mathbf{i}} R$ for the *projection* of R on $\mathbf{i}$, the relation $\{(a_{i_1}, \dots, a_{i_k}) : (a_1, \dots, a_n) \in R$ for some values of the a_j, $j \notin \mathbf{i}\}$. For $I \subseteq [n]$, we write $\mathsf{pr}_I R$ as shorthand for $\mathsf{pr}_{\mathbf{i}} R$, where $\mathbf{i}$ is the enumeration of I's elements in increasing order. For the relation $\{\mathbf{t}\}$, where $\mathbf{t}$ is a single n-tuple, we write $\mathsf{pr}_{\mathbf{i}} \mathbf{t}$ rather than $\mathsf{pr}_{\mathbf{i}} \{\mathbf{t}\}$.

We define the *size* of a relation H as $\|H\| = \ell r$, the number of elements in its matrix, and the size of Γ as $\|\Gamma\| = \sum_{H \in \Gamma} \|H\|$. To avoid trivialities, we will assume that every relation $H \in \Gamma$ is nonempty, i.e. that $\|H\| > 0$. We will also assume that every $d \in D$ appears in a tuple of some relation $H \in \Gamma$. If this is not so for some d, we can remove it from D. It then follows that $\|\Gamma\| \geq q$.

Let $V = \{\nu_1, \nu_2, \dots, \nu_n\}$ be a finite set of *variables*. An *assignment* is a function $\mathbf{x}\colon V \to D$. We will abbreviate $\mathbf{x}(\nu_i)$ to x_i. If $\{i_1, i_2, \dots, i_r\} \subseteq [n]$, we write $H(x_{i_1}, x_{i_2}, \dots, x_{i_r})$ for the relation $\Theta = \{\mathbf{x} : (x_{i_1}, x_{i_2}, \dots, x_{i_r}) \in H\}$ and we refer to this as a *constraint*. Then $(\nu_{i_1}, \nu_{i_2}, \dots, \nu_{i_r})$ is the *scope* of the constraint and we say that $\mathbf{x}$ is a *satisfying* assignment for the constraint if $\mathbf{x} \in \Theta$.

A Γ*-formula* Φ in a set of variables $\{x_1, x_2, \dots, x_n\}$ is a conjunction of constraints $\Theta_1 \wedge \dots \wedge \Theta_m$. We will identify the variables with the x_i above, although strictly they are only a *model* of the formula. Note that the precise labelling of the variables in Φ has no real significance. A formula remains the same if its variables are relabelled under a bijection to any other set of variable names.

A Γ-formula Φ describes an instance of the *constraint satisfaction problem* (CSP) with *constraint language* Γ. A satisfying assignment for Φ is an assignment that satisfies all Θ_i ($i \in [m]$). The set of all satisfying assignments for Φ is the Γ*-definable* relation R_Φ over D. We make no distinction between Φ and R_Φ, unless this could cause confusion.

2.2. Definability. A *primitive positive* (pp) formula Ψ is a Γ-formula Φ with existential quantification over some subset of the variables. A satisfying assignment for Ψ is any satisfying assignment for Φ. The unquantified (free) variables then determine the *pp-definable* relation R_Ψ, a projection of R_Φ. Note that any permutation of the columns of a pp-definable relation is, itself, pp-definable. Again, we make no distinction between Ψ and R_Ψ.

The set of all Γ-definable relations is denoted by $\mathsf{CSP}(\Gamma)$ and the set of all pp-definable relations is the *relational clone* $\langle\Gamma\rangle$. If $\Gamma = \{H, =\}$, we just write $\langle H\rangle$. An *equivalence relation* in $\langle\Gamma\rangle$ is called a *congruence.*

2.3. Polymorphisms. A k-ary *polymorphism* of Γ is any function $\psi\colon D^k \to D$, for some k, that preserves all the relations in Γ. By this we mean that, for every r-ary relation $H \in \Gamma$ and every sequence $\mathbf{u}_1, \dots, \mathbf{u}_k$ of r-tuples in H,

$$\psi(\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k) = \big(\psi(u_{1,1}, \dots, u_{k,1}), \psi(u_{1,2}, \dots, u_{k,2}), \dots, \psi(u_{1,r}, \dots, u_{k,r})\big) \in H\,.$$

It is well known that any polymorphism of Γ preserves all relations in $\langle\Gamma\rangle$ (see Lemma 4).

A *Mal'tsev polymorphism* of Γ is a polymorphism $\varphi\colon D^3 \to D$ such that, for all $a, b \in D$, $\varphi(a, b, b) = \varphi(b, b, a) = a$. (So, in particular, $\varphi(a, a, a) = a$.) We will usually present calculations using φ in a four-row table. The first three rows give the triple of "input" tuples $\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3$ and the fourth gives the "output" $\varphi(\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3)$. For example, the table below indicates that $\varphi(a\mathbf{u}, a\mathbf{v}, b\mathbf{w}) = (b, \varphi(\mathbf{u}, \mathbf{v}, \mathbf{w}))$.

$$\begin{array}{cc} a & \mathbf{u} \\ a & \mathbf{v} \\ b & \mathbf{w} \\ \hline b & \varphi(\mathbf{u}, \mathbf{v}, \mathbf{w})\,. \end{array}$$

2.4. Complexity. For any alphabet Σ, we denote by FP the class of functions $f\colon \Sigma^* \to \mathbb{N}$ for which there is a deterministic, polynomial-time Turing machine that, given input $x \in \Sigma^*$, writes $f(x)$ (in binary) to its output tape. $\#\mathsf{P}$ is the class of functions $f\colon \Sigma^* \to \mathbb{N}$ such that there is a nondeterministic, polynomial-time Turing machine that has exactly $f(x)$ accepting computations for every input $x \in \Sigma^*$.

Completeness for $\#\mathsf{P}$ is defined with respect to polynomial-time Turing reductions [29], also known as *Cook reductions.* For functions $f, g\colon \Sigma^* \to \mathbb{N}$, a *polynomial-time Turing reduction* from f to g is a polynomial-time oracle Turing machine that can compute f using an oracle for g. A function $f \in \#\mathsf{P}$ is $\#\mathsf{P}$-*complete* if there is a Cook reduction to f from every problem in $\#\mathsf{P}$.

The class $\#\mathsf{P}$ plays a role in the complexity of counting problems analogous to that played by NP in decision problems. Note, however, that, subject to standard complexity-theoretic assumptions, $\#\mathsf{P}$-complete problems are much harder than NP-complete problems. Toda has shown that $\mathsf{P}^{\#\mathsf{P}}$ includes the whole of the polynomial-time hierarchy [27], whereas P^{NP} is just the hierarchy's second level.

3. Rectangular relations. A binary relation $B \subseteq A_1 \times A_2$ is called *rectangular* if $(a, c), (a, d), (b, c) \in B$ implies $(b, d) \in B$ for all $a, b \in A_1$, $c, d \in A_2$. We may view B as an undirected bipartite graph $\mathcal{G}_B$, with vertex bipartition A_1, A_2 and edge set $E_B = \{\{a_1, a_2\} : (a_1, a_2) \in B\}$. Note that we do not insist that $A_1 \cap A_2 = \emptyset$ but, if $a \in A_1 \cap A_2$, a is regarded as labelling two distinct vertices, one in A_1 and one in A_2. Formally, A_1 and A_2 should be replaced by the disjoint vertex sets $\{1\} \times A_1$ and $\{2\} \times A_2$ but this would unduly complicate the notation. We will assume that $\mathsf{pr}_i B = A_i$ $(i = 1, 2)$, so that $\mathcal{G}_B$ has no isolated vertices. The connected components of $\mathcal{G}_B$ will be called the *blocks* of B.

Rectangular relations have very simple structure.

Lemma 1. *If B is rectangular, $\mathcal{G}_B$ comprises k bipartite cliques, for some $k \leq \min\{|A_1|, |A_2|\}$.*

Proof. Let k be the number of connected components of $\mathcal{G}_B$. Every vertex is included in an edge so $k \leq \min\{|A_1|, |A_2|\}$. Consider any component C and suppose it is not a bipartite clique. Let $a \in A_1 \cap C$, $z \in A_2 \cap C$ be such that $\{a, z\} \notin E_B$.

Thus, a shortest path in C from a to z has length at least 3. If a, b, c, d are the first four vertices on such a path, then $\{a,b\},\{b,c\},\{c,d\} \in E_B$, but $\{a,d\} \notin E_B$ as, otherwise, there would be a shorter path from a to z. But this is equivalent to $(a,b),(c,b),(c,d) \in B$ and $(a,d) \notin B$, contradicting rectangularity. $\square$

Where appropriate, we do not distinguish between B and $\mathcal{G}_B$. For example, we will refer to a connected component of $\mathcal{G}_B$ as a block.

Corollary 2. *The relations*

$$\theta_1(x_1,x_2) \equiv \exists y\,\big(B(x_1,y) \wedge B(x_2,y)\big) \quad \textit{and} \quad \theta_2(y_1,y_2) \equiv \exists x\,\big(B(x,y_1) \wedge B(x,y_2)\big)$$

are equivalence relations on $\mathsf{pr}_1 B$, $\mathsf{pr}_2 B$ *respectively. The equivalence classes of* θ_1 *and* θ_2 *are in one-to-one correspondence.*

Proof. The blocks of B induce partitions of A_1 and A_2 which are in one-to-one correspondence. These clearly define the equivalence classes of θ_1 and θ_2. $\square$

Corollary 3. *If* Γ *is a constraint language and* $B \in \langle\Gamma\rangle$ *is rectangular, then the relations* θ_1 *and* θ_2 *of Corollary 2 are congruences in* $\langle\Gamma\rangle$.

Proof. Since B has a pp-definition, so too do θ_1 and θ_2. $\square$

We say that a relation $R \subseteq D^n$ for $n \geq 2$ is rectangular if every expression of R as a binary relation in $D^k \times D^{n-k}$ $(1 \leq k < n)$ is rectangular. We call a constraint language Γ *strongly rectangular* if every relation $B \in \langle\Gamma\rangle$ of arity at least 2 is rectangular. If $R \subseteq D^n$ is a relation, we say that it is strongly rectangular if $\langle R\rangle$ is strongly rectangular. If $R \in \langle\Gamma\rangle$ for a strongly rectangular Γ, then R is strongly rectangular, since $\langle R\rangle \subseteq \langle\Gamma\rangle$.

From the definition, it is not clear whether the strong rectangularity of Γ is even decidable, since $\langle\Gamma\rangle$ is an infinite set. However, it is decidable, as we will now show. The following result is usually proven in an algebraic setting. That proof is not difficult, but requires an understanding of concepts from universal algebra, such as *free algebras* and *varieties* [12]. Therefore, we will give a proof in the relational setting. Moreover, we believe that this proof will provide rather more insight for the reader whose primary interest is in relations.

First, we require the following lemma, which is well-known from the folklore; we provide a proof for completeness.

Lemma 4. φ *is a polymorphism of* Γ *if, and only if, it is a polymorphism of* $\langle\Gamma\rangle$.

Proof. Let φ be a polymorphism of Γ and let $R \in \langle\Gamma\rangle$. We prove that φ is a polymorphism of R by induction on the structure of the defining formula of R. The base case, atomic formulae ($H(\mathbf{x})$ for relations $H \in \Gamma$) is trivial.

Suppose R is defined by $\exists y\,\psi(\mathbf{x},y)$. If $\mathbf{a}_1,\mathbf{a}_2,\mathbf{a}_3 \in R$, then there are b_1,b_2,b_3 such that $\mathbf{a}_i b_i \in \psi$ $(i \in \{1,2,3\})$. If φ is a polymorphism of ψ, then it follows that $\mathbf{c}d = \varphi(\mathbf{a}_1 b_1, \mathbf{a}_2 b_2, \mathbf{a}_3 b_3) \in \psi$, which means that $\mathbf{c} \in R$, as required.

Finally, suppose R is defined by $\psi(\mathbf{x}) \wedge \chi(\mathbf{x})$. If $\mathbf{a}_1,\mathbf{a}_2,\mathbf{a}_3 \in R$, then $\mathbf{a}_i \in \psi \cap \chi$ for each i. If φ is a polymorphism of ψ and of χ then $\mathbf{c} = \varphi(\mathbf{a}_1,\mathbf{a}_2,\mathbf{a}_3) \in \psi \cap \chi$ and, therefore, $\mathbf{c} \in R$, as required.

Conversely, $\Gamma \subseteq \langle\Gamma\rangle$ so every polymorphism of $\langle\Gamma\rangle$ is a polymorphism of Γ. $\square$

Lemma 5. *A constraint langauge* Γ *is strongly rectangular if, and only if, it has a Mal'tsev polymorphism.*

Proof. Suppose Γ has a Mal'tsev polymorphism φ. Consider any pp-definable binary relation $B \subseteq D^r \times D^s$. By Lemma 4, φ is also a polymorphism of B. If $(\mathbf{a},\mathbf{c}),(\mathbf{a},\mathbf{d}),(\mathbf{b},\mathbf{d}) \in B$ then we have $(\varphi(\mathbf{a},\mathbf{a},\mathbf{b}),\varphi(\mathbf{c},\mathbf{d},\mathbf{d})) = (\mathbf{b},\mathbf{c}) \in B$, from the definition of a Mal'tsev polymorphism. Thus, B is rectangular and hence Γ is strongly rectangular.

Conversely, suppose Γ is strongly rectangular. Denote the relation $H \in \Gamma$ by $H = \{\mathbf{u}_i^H : i \in [\ell_H]\}$, where $\mathbf{u}_i^H \in D^{r_H}$. Consider the Γ-formula

$$\Phi(\mathbf{x}) \;=\; \bigwedge_{H\in\Gamma} \; \bigwedge_{i_1\in[\ell_H]} \; \bigwedge_{i_2\in[\ell_H]} \; \bigwedge_{i_3\in[\ell_H]} H\big(\mathbf{x}^H_{i_1,i_2,i_3}\big)\,,$$

where $\mathbf{x}^H_{i_1,i_2,i_3}$ is an r_H-tuple of variables, distinct for all $H \in \Gamma$, $i_1, i_2, i_3 \in [\ell_H]$. Thus, the relation R_Φ has arity $r_\Phi = \sum_{H\in\Gamma} r_H \ell_H^3$ and $|R_\Phi| = \prod_{H\in\Gamma} \ell_H{}^{\ell_H^3}$.

Clearly R_Φ has three tuples $\mathbf{u}_1$, $\mathbf{u}_2$, $\mathbf{u}_3$ such that the sub-tuple of $\mathbf{u}_j$ corresponding to $\mathbf{x}^H_{i_1,i_2,i_3}$ is $\mathbf{u}^H_{i_j}$ for each $j \in \{1,2,3\}$ and each $i_1, i_2, i_3 \in [\ell_H]$. Then $U = \{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ has the following universality property for Γ. For all $H \in \Gamma$ and every triple of (not necessarily distinct) tuples $\mathbf{t}_1$, $\mathbf{t}_2$, $\mathbf{t}_3 \in H$, there is a set $I = I(\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3)$ with $I \subseteq [r_\Phi]$, $|I| = r_H$ such that $\mathsf{pr}_I R_\Phi = H$ and $\mathsf{pr}_I \mathbf{u}_i = \mathbf{t}_i$ $(i = 1, 2, 3)$.

Now, for each set of identical columns in U, we impose equality on the corresponding variables in Φ, to give a Γ-formula Φ'. Let U' be the resulting submatrix of U, with rows $\mathbf{u}_1'$, $\mathbf{u}_2'$, $\mathbf{u}_3'$. Observe that U' is obtained by deleting copies of columns in U. Therefore U' has no identical columns and has a column (a, b, c) for all $a, b, c \in \mathsf{pr}_k H$ with $H \in \Gamma$ and $k \in [r_H]$.

Next, for all columns (a, b, c) of U' such that $b \notin \{a, c\}$, we impose existential quantification on the corresponding variables in Φ', to give a pp-formula Φ''. Let U'' be the submatrix of U' with rows $\mathbf{u}_1''$, $\mathbf{u}_2''$, $\mathbf{u}_3''$ corresponding to $\mathbf{u}_1'$, $\mathbf{u}_2'$, $\mathbf{u}_3'$. Then U'' results from deleting columns in U' and U'' has columns of the form (a, a, b) or (c, d, d). Thus, after rearranging columns (relabelling variables), we will have

$$U'' \;=\; \begin{bmatrix} \mathbf{u}_1'' \\ \mathbf{u}_2'' \\ \mathbf{u}_3'' \end{bmatrix} \;=\; \begin{bmatrix} \mathbf{a} & \mathbf{c} \\ \mathbf{a} & \mathbf{d} \\ \mathbf{b} & \mathbf{d} \end{bmatrix}\,,$$

for some nonempty tuples $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, $\mathbf{d}$. By strong rectangularity, this implies that $\mathbf{u}'' = \begin{bmatrix} \mathbf{b} & \mathbf{c} \end{bmatrix} \in R_{\Phi''}$.

Removing the existential quantification in Φ'', $\mathbf{u}''$ can be extended to $\mathbf{u}' \in R_{\Phi'}$. Now, if column k of U' is (a, b, c) say, we define $\varphi(a, b, c) = u_k'$. This is unambiguous, since U' has no identical columns. Thus, $\mathbf{u}' = \varphi(\mathbf{u}_1', \mathbf{u}_2', \mathbf{u}_3') \in R_{\Phi'}$. If, for any $a, b, c \in D$, $\varphi(a, b, c)$ remains undefined, we will set $\varphi(a, b, c) = a$ unless $a = b$, in which case $\varphi(a, b, c) = c$. Clearly φ satisfies $\varphi(a, b, b) = \varphi(b, b, a) = a$, for all $a, b \in D$, and so has the Mal'tsev property.

Removing the equalities between variables in Φ', $\mathbf{u}'$ can be further extended to $\mathbf{u} = \varphi(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3) \in R_\Phi$. This is consistent since $\mathbf{u}$ satisfies the equalities imposed on Φ to give Φ'. Now, for any $\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3 \in H$, the universality property of U implies that, for some I, $\mathsf{pr}_I \mathbf{u} = \varphi(\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3) \in H$. Thus, φ preserves all $H \in \Gamma$, so it is a polymorphism and hence a Mal'tsev polymorphism. □

Remark 1. Observe that the proof of Lemma 5 uses all the elements of pp-definability. Thus, if Lemma 5 is to hold true, the definition of strong rectangularity cannot be significantly weakened.

Remark 2. The proof of Lemma 5 is constructive and, hence, implies an algorithm for deciding whether Γ is strongly rectangular and, if so, determining a Mal'tsev polymorphism φ. However, we describe a more efficient method in Lemma 8 below.

Note that strong rectangularity is invariant under permutations of the columns of a relation, both by Lemma 5 (since permutations of columns do not affect Mal'tsev polymorphisms) and by the fact that permutations are pp-definable. We will use this fact repeatedly and consider a relation $R \subseteq D^n$ for some $n > 2$ to be a binary relation on $D^k \times D^{n-k}$ or a ternary relation on $D^k \times D^\ell \times D^{n-k-\ell}$, for any appropriate values of k and ℓ.

In the algebraic setting, the result corresponding to Lemma 5 is that $\langle\Gamma\rangle$ has a Mal'tsev polymorphism if, and only if, Γ is *congruence permutable*. See, for example, [12]. This has the following meaning. If ρ_1 and ρ_2 are congruences on a pp-definable set $A \subseteq D^r$, define the *relational product* $\psi = \rho_1 \circ \rho_2$ by $\psi(\mathbf{x}, \mathbf{y}) = \exists\mathbf{z}\,\big(\chi(\mathbf{z}) \wedge \rho_1(\mathbf{x}, \mathbf{z}) \wedge \rho_2(\mathbf{z}, \mathbf{y})\big)$, where χ is the formula defining A. Then ρ_1, ρ_2 are *permutable* if $\psi(\mathbf{u}, \mathbf{v})$ implies $\psi(\mathbf{v}, \mathbf{u})$ for all $\mathbf{u}, \mathbf{v} \in A$ or, equivalently, $\rho_1 \circ \rho_2 = \rho_2 \circ \rho_1$. Now Γ is congruence permutable if every pair of congruences on the same set A is permutable. For completeness, we will prove the following.

Lemma 6. *Γ is strongly rectangular if, and only if, it is congruence permutable.*

Proof. Suppose Γ is strongly rectangular. If ρ_1, ρ_2 are congruences on a pp-definable set $A \subseteq D^r$, let ψ be the relational product, as defined above. Clearly ψ is a pp-definable binary relation on D^r. Then, if $(\mathbf{u}, \mathbf{v}) \in \psi$, we have $(\mathbf{u}, \mathbf{u})$, $(\mathbf{u}, \mathbf{v})$, $(\mathbf{v}, \mathbf{v}) \in \psi$, since ρ_1 and ρ_2 are congruences. But this implies $(\mathbf{v}, \mathbf{u}) \in \psi$ since ψ is rectangular. Thus, Γ is congruence permutable.

Conversely, if Γ is congruence permutable, consider a pp-definable relation $B \subseteq D^r \times D^s$. Define a relation $\sim_1$ on B by $(\mathbf{x}_1, \mathbf{y}_1) \sim_1 (\mathbf{x}_2, \mathbf{y}_2)$ if, and only if, $(\mathbf{x}_1, \mathbf{y}_1) \in B$, $(\mathbf{x}_2, \mathbf{y}_2) \in B$ and $\mathbf{x}_1 = \mathbf{x}_2$. This is pp-definable, by $B(\mathbf{x}_1, \mathbf{y}_1) \wedge B(\mathbf{x}_2, \mathbf{y}_2) \wedge (\mathbf{x}_1 = \mathbf{x}_2)$, and is clearly an equivalence relation. Hence it is a congruence. Similarly, define a congruence $\sim_2$ on D^{r+s} by $(\mathbf{x}_1, \mathbf{y}_1) \sim_2 (\mathbf{x}_2, \mathbf{y}_2)$ if, and only if, $(\mathbf{x}_1, \mathbf{y}_1)$, $(\mathbf{x}_2, \mathbf{y}_2) \in B$ and $\mathbf{y}_1 = \mathbf{y}_2$. Let $\psi = \sim_1 \circ \sim_2$.

Suppose $\big((\mathbf{a}, \mathbf{c}), (\mathbf{b}, \mathbf{d})\big) \in \psi$. Then there exists $(\mathbf{u}, \mathbf{v}) \in B$ such that $(\mathbf{a}, \mathbf{c}) \sim_1 (\mathbf{u}, \mathbf{v}) \sim_2 (\mathbf{b}, \mathbf{d})$. Thus, $(\mathbf{u}, \mathbf{v}) = (\mathbf{a}, \mathbf{d})$ and, hence, $(\mathbf{a}, \mathbf{c})$, $(\mathbf{a}, \mathbf{d})$, $(\mathbf{b}, \mathbf{d}) \in B$. Congruence permutability implies $\big((\mathbf{b}, \mathbf{d}), (\mathbf{a}, \mathbf{c})\big) \in \psi$. Hence there exists $(\mathbf{u}', \mathbf{v}') \in B$ such that $(\mathbf{b}, \mathbf{d}) \sim_1 (\mathbf{u}', \mathbf{v}') \sim_2 (\mathbf{a}, \mathbf{c})$. Thus, $(\mathbf{u}', \mathbf{v}') = (\mathbf{b}, \mathbf{c})$. Therefore we have $(\mathbf{b}, \mathbf{c}) \in B$ and Γ is strongly rectangular. □

Corollary 7. *Γ is congruence permutable if, and only if, it has a Mal'tsev polymorphism.*

Proof. This follows directly from Lemmas 5 and 6. □

We will now consider the complexity of deciding whether Γ is strongly rectangular.

Lemma 8. *We can decide whether Γ is strongly rectangular in $\mathcal{O}(\|\Gamma\|^4)$ time and, if so, determine a Mal'tsev polymorphism φ.*

Proof. Observe that there are at most $q^{q(q-1)^2}$ possible Mal'tsev operations $D^3 \to D$. This follows since there are $q(q-1)^2$ triples $a, b, c \in D$ which have $b \notin \{a, c\}$. For all other triples, the value of $\varphi(a, b, c)$ is determined by the condition that φ is Mal'tsev. Thus, there are $\mathcal{O}(1)$ possibilities for φ. For an r-ary relation $H \in \Gamma$ with ℓ tuples, we can check in $\mathcal{O}(\ell^4 r) = \mathcal{O}(\|H\|^4)$ time whether H is preserved by any of them. If so, we have φ; if not, Γ is not strongly rectangular. □

Remark 3. We have assumed that q is a constant in Lemma 8. We revisit this question in Section 8, where we make no such assumption.

In view of Lemma 8, we may assume that we have determined a Mal'tsev polymorphism φ for any given strongly rectangular Γ.

Strongly rectangular constraint languages have another useful property. For each $a \in D$, define the *constant relation* $\chi_a = \{(a)\}$. Then the constraint $\chi_a(x_i)$ fixes the value of x_i to be a.

Lemma 9. *If Γ is strongly rectangular, then so is $\Gamma' = \Gamma \cup \{\chi_a\}$.*

Proof. By Lemma 5, Γ is preserved by a Mal'tsev polymorphism φ. Since

$\varphi(a,a,a)=a$ for any $a\in D$, φ also preserves χ_a. Thus φ preserves Γ', so Γ' is strongly rectangular, by Lemma 5. ∎

In the light of Lemma 9, we may assume that $\{\chi_a : a\in D\}\subseteq\Gamma$ whenever Γ is strongly rectangular.

Remark 4. More generally, the property of a polymorphism ψ that we have used in Lemma 9, that $\psi(x,x,\ldots,x)=x$ for any $x\in D$, is called *idempotence* in the algebraic literature on CSP.

4. The structure of strongly rectangular relations. Let $R\subseteq D^n$ be a strongly rectangular relation. For any $i\in[n]$, we say that an n-tuple $\mathbf{t}\in R$ is a *witness* for $a\in \mathsf{pr}_i R$ if $t_i=a$. We will abbreviate this by saying that $\mathbf{t}$ witnesses (a,i). If $\mathbf{t}=(\mathbf{u},a,\mathbf{v})\in R$, we call $\mathbf{u}$ a *prefix* for a. Now define a relation $\sim_i$ on $\mathsf{pr}_i R$ by $a\sim_i b$ if, and only if, there exists $\mathbf{u}\in D^{i-1}$ which is a common prefix for a and b. That is, there exist $\mathbf{v}_a,\mathbf{v}_b\in D^{n-i}$ such that $(\mathbf{u},a,\mathbf{v}_a)$, $(\mathbf{u},b,\mathbf{v}_b)\in R$.

Lemma 10. *$\sim_i$ is an equivalence relation on $\mathsf{pr}_i R$ and a congruence in $\langle R\rangle$.*

Proof. Consider the binary relation B on $\mathsf{pr}_{[i-1]}R\times\mathsf{pr}_i R$ defined by $B(\mathbf{u},a)=\exists\mathbf{y}\,R(\mathbf{u},a,\mathbf{y})$. Then $\sim_i$ is the equivalence relation θ_2 of Corollary 2, which is a congruence by Corollary 3. ∎

Let $\mathcal{E}_{i,k}$ $(k\in[\kappa_i])$ be the equivalence classes of $\sim_i$ for $\kappa_i\in[q]$, $i\in[n]$. Observe that $\kappa_1=1$, since all $a\in\mathsf{pr}_1R$ have witnesses with the common empty prefix. More generally, we make the following observation, which follows directly from the block structure of the relation B in the proof of Lemma 10.

Corollary 11. *There is a common prefix $\mathbf{u}_{i,k}\in D^{i-1}$ for all $a\in\mathcal{E}_{i,k}$ ($k\in[\kappa_i], i\in[n]$) and we can choose $\mathbf{u}_{i,k}$ to be any prefix of any $a\in\mathcal{E}_{i,k}$.*

Following Bulatov and Dalmau [4], if H is any relation and φ a Mal'tsev operation (i.e., a ternary function that is not necessarily a polymorphism but has the property that $\varphi(a,b,b)=\varphi(b,b,a)=a$ for all $a,b\in D$), then $\mathsf{cl}_\varphi H$ is the smallest relation that contains H and which is closed under φ. Clearly $\mathsf{cl}_\varphi H$ is a strongly rectangular relation with polymorphism φ and we say that the H *generates* $\mathsf{cl}_\varphi H$. The following observation, from [4], gives a simple but important fact.

Lemma 12. *Let H be an n-ary relation. If $I\subseteq[n]$, then $\mathsf{cl}_\varphi\mathsf{pr}_I H=\mathsf{pr}_I\mathsf{cl}_\varphi H$.*

Proof. Consider generating $\mathsf{cl}_\varphi\mathsf{pr}_I H$ while retaining all n columns of H. Each row of the resulting n-ary relation will be in $\mathsf{cl}_\varphi H$, so we have $\mathsf{cl}_\varphi\mathsf{pr}_I H\subseteq\mathsf{pr}_I\mathsf{cl}_\varphi H$. But further operations to generate $\mathsf{cl}_\varphi H$ cannot add new rows to $\mathsf{cl}_\varphi\mathsf{pr}_I H$. So, in fact, we have $\mathsf{cl}_\varphi\mathsf{pr}_I H=\mathsf{pr}_I\mathsf{cl}_\varphi H$. ∎

Let $S=\{\mathbf{t}_1,\mathbf{t}_2,\ldots,\mathbf{t}_s\}$ be a set of n-tuples, presented as an $s\times n$ matrix. If $I\subseteq[n]$, we will need to compute a relation $T\subseteq\mathsf{cl}_\varphi S$ such that $\mathsf{pr}_I T=\mathsf{cl}_\varphi\mathsf{pr}_I S=\mathsf{pr}_I\mathsf{cl}_\varphi S$.

Lemma 13. *If $\ell=|\mathsf{pr}_I\mathsf{cl}_\varphi S|$ and $s=|S|$, then a relation $T\subseteq\mathsf{cl}_\varphi S$ such that $\mathsf{pr}_I T=\mathsf{pr}_I\mathsf{cl}_\varphi S$ can be computed in time $\mathcal{O}(n\ell^3+s\ell^4)$.*

Proof. Consider the algorithm Closure, on the following page.

The correctness of Closure is trivial. At termination, all ℓ^3 triples (k_1,k_2,k_3) with $k_1,k_2,k_3\in[\ell]$ have been considered for generating new n-tuples (in line 6), so we have computed $\mathsf{cl}_\varphi\mathsf{pr}_I S$. The analysis is equally easy. There are ℓ^3 triples (k_1,k_2,k_3). For each triple, the generation in line 6 takes $\mathcal{O}(n)$ time and the search in line 7 requires $\mathcal{O}(s\ell)$ time, with the obvious implementations. Thus, the total is $\mathcal{O}(n\ell^3+s\ell^4)$. ∎

The procedure outlined in [4] has complexity $\mathcal{O}(n\ell^4+s\ell^5)$, since the same triple (k_1,k_2,k_3) can appear $\Omega(\ell)$ times. The procedure Closure simply avoids this.

The time complexity of Closure could be improved, for example, by using a more sophisticated data structure to implement the searches in line 7. However we

```
procedure Closure($I$)
  $\ell \leftarrow s$, $j_1 \leftarrow 2$
  while $j_1 \le \ell$ do
      for $j_2 \in [j_1]$ do
          for $j_3 \in [j_2]$ do
              for all permutations $(k_1, k_2, k_3)$ of $\{j_1, j_2, j_3\}$ such that $k_2 \notin \{k_1, k_3\}$ do
                  $\mathbf{u} \leftarrow \varphi(\mathbf{t}_{k_1}, \mathbf{t}_{k_2}, \mathbf{t}_{k_3})$
                  if there is no $j \in [\ell]$ such that $\mathsf{pr}_I \mathbf{t}_j = \mathsf{pr}_I \mathbf{u}$ then
                      $\ell \leftarrow \ell + 1$, $\mathbf{t}_\ell \leftarrow \mathbf{u}$
      $j_1 \leftarrow j_1 + 1$
```

do not pursue such issues here, or elsewhere in the paper.

Now we define a *frame* for an n-ary relation R to be a set $F \subseteq R$ such that

(a) $\mathsf{pr}_i F = \mathsf{pr}_i R$ for each $i \in [n]$; and

(b) there is a $\mathbf{v}_{i,k} \in D^{i-1}$ for each equivalence class $\mathcal{E}_{i,k}$ of $\sim_i$ ($k \in [\kappa_i], i \in [n]$) such that, for each $a \in \mathcal{E}_{i,k}$, there exists a $\mathbf{w}_a \in F$ with $\mathsf{pr}_{[i]} \mathbf{w}_a = \mathbf{v}_{i,k} a$.

Clearly, R itself satisfies the definition of a frame, so every relation has at least one frame. However, we will show that strongly rectangular relations have frames that can be much smaller than R and we call a frame for a strongly rectangular relation $R \subseteq D^n$ *small* if $|F| \le n(q-1)+1$.

A *witness function* for a frame F of the relation R is a function $\boldsymbol{\omega}\colon D \times [n] \to F$ such that $\boldsymbol{\omega}(a,i)$ witnesses (a,i) for all $a \in \mathsf{pr}_i R$ and $i \in [n]$ and $\mathsf{pr}_{[i-1]} \boldsymbol{\omega}(a,i) = \mathsf{pr}_{[i-1]} \boldsymbol{\omega}(b,i)$ when $a \sim_i b$. That is, $\boldsymbol{\omega}(a,i)$ returns a witness for (a,i) and, if (a_1,i), $\dots$, (a_k,i) have witnesses with a common prefix, then $\boldsymbol{\omega}$ returns such witnesses.

Lemma 14. *Let F be a frame for a strongly rectangular relation $R \subseteq D^n$. We can determine a small frame F' for R and a surjective witness function $\boldsymbol{\omega}'\colon D \times [n] \to F'$ in time $\mathcal{O}(\|F\|^2)$.*

Proof. In time $\mathcal{O}(\|F\|)^2$, we can compute the relations $\sim_i$ ($i \in [n]$) and common prefixes for each $\sim_i$-equivalence class. Hence, we can compute a witness function $\boldsymbol{\omega}$ for F. Further, we may delete from F any tuple $\mathbf{t}$ for which $\boldsymbol{\omega}^{-1}(\mathbf{t}) = \emptyset$. Because $\boldsymbol{\omega}$ is a witness function, the resulting set is still a frame for R and has size at most $\sum_{i \in [n]} |\mathsf{pr}_i R| \le nq$.

Now we construct F' and $\boldsymbol{\omega}'$ as follows. Choose any $\mathbf{f} \in F$ and set $F' = \{\mathbf{f}\}$. Then, for each $i \in [n]$, do the following. Let $g = \boldsymbol{\omega}(f_i, i)$ and set $\boldsymbol{\omega}'(f_i, i) \leftarrow \mathbf{f}$. Now, consider in turn each $a \ne f_i$ such that $a \sim_i f_i$ and let $\mathbf{h} = \boldsymbol{\omega}(a,i)$. Note that $\mathbf{g}$ and $\mathbf{h}$ have the same prefix $\mathbf{u}' \in D^{i-1}$, since F is a frame, and suppose $\mathbf{f}$ has prefix $\mathbf{u} \in D^{i-1}$. Then set $\mathbf{h}' \leftarrow \varphi(\mathbf{f}, \mathbf{g}, \mathbf{h})$, $F' \leftarrow F' \cup \{\mathbf{h}'\}$ and $\boldsymbol{\omega}'(a,i) \leftarrow \mathbf{h}'$. Since

$$
\begin{array}{cccccc}
\mathbf{f} & : & \mathbf{u} & f_i & \mathbf{v} \\
\mathbf{g} & : & \mathbf{u}' & f_i & \mathbf{v}' \\
\mathbf{h} & : & \mathbf{u}' & a & \mathbf{v}_a \\
\hline
\mathbf{h}' & : & \mathbf{u} & a & \varphi(\mathbf{v}, \mathbf{v}', \mathbf{v}_a)\,,
\end{array}
$$

this ensures that F' retains property (b) of a frame. Having performed these steps for each $i \in [n]$, we deal with those $a \in \mathsf{pr}_i F$ with $a \not\sim_i f_i$ by setting $F' \leftarrow F' \cup \{\boldsymbol{\omega}(a,i)\}$ and $\boldsymbol{\omega}'(a,i) \leftarrow \boldsymbol{\omega}(a,i)$.

The final size of F' can be bounded as follows. The tuple $\mathbf{f}$ witnesses (f_i, i) for all $i \in [n]$. Then, for each $i \in [n]$, there is at most one tuple in F' witnessing (a,i) for each $a \in \mathsf{pr}_i R \setminus \{f_i\}$. Since there are, in total, $\sum_{i=1}^n \bigl(|\mathsf{pr}_i R| - 1\bigr) \le n(q-1)$ such pairs (a,i), it follows that F' is a small frame.

The time bound is easy. Given the function $\boldsymbol{\omega}$, we can determine the $\mathbf{h}'$ in $\mathcal{O}(n)$ for each $i \in [n]$. All other operations require $\mathcal{O}(1)$ time for each $i \in [n]$. Thus, we can need only $\mathcal{O}(n^2) = \mathcal{O}(\|F\|^2)$ time once we have determined $\boldsymbol{\omega}$, which can also be done in $\mathcal{O}(\|F\|^2)$ time. □

Remark 5. The upper bound for the size of a small frame is achieved by the complete relation D^n. We exhibit a small frame for D^n in Lemma 18 below. However, a frame can be much smaller than this upper bound $n(q-1)+1$. Consider, for example, the n-ary relation $R = \{(a, \dots, a) : a \in D\}$. It is easy to show that R is strongly rectangular. However, it is also easy to see that $F = R$ is a frame, with $\boldsymbol{\omega}(a, i) = (a, \dots, a)$ $(i \in [n])$ and $|F| = q$.

Remark 6. The *compact representations* of Bulatov and Dalmau [4] are not necessarily frames and can have size $nq^2/2$. However, it appears that a frame could be constructed efficiently from such a representation using methods similar to those of Lemma 14.

We will suppose below that all frames are small. If necessary, this can be achieved using Lemma 14. Note that we do not assume that a frame for R can actually generate R, since this is entailed by the following.

Lemma 15. *If R is strongly rectangular with Mal'tsev polymorphism φ and F is a frame for R, then $\mathsf{cl}_\varphi F = R$.*

Proof. $F \subseteq R$ so $\mathsf{cl}_\varphi F \subseteq \mathsf{cl}_\varphi R = R$. It remains to show that $R \subseteq \mathsf{cl}_\varphi F$.

We show by induction on $i \in [n]$ that $\mathsf{pr}_{[i]} R \subseteq \mathsf{pr}_{[i]} \mathsf{cl}_\varphi F$. The base case, $i = 1$, is trivial as $\mathsf{pr}_1 R = \mathsf{pr}_1 F$ by definition. Suppose that $\mathsf{pr}_{[i-1]} R \subseteq \mathsf{pr}_{[i-1]} \mathsf{cl}_\varphi F$ and let $\mathbf{t} = (t_1, \dots, t_n) = (\mathbf{u}, t_i, \mathbf{v}) \in R$. By the inductive hypothesis, we have $\mathbf{u} \in \mathsf{pr}_{[i-1]} \mathsf{cl}_\varphi F$ so there is a tuple $\mathbf{t}' = (\mathbf{u}, t_i', \mathbf{v}') \in \mathsf{cl}_\varphi F \subseteq R$. Therefore, $t_i' \sim_i t_i$, which means there are tuples $(\mathbf{u}', t_i, \mathbf{w})$ and $(\mathbf{u}', t_i', \mathbf{w}')$ in F witnessing (t_i, i) and (t_i', i), respectively. Thus, we have

$$\begin{array}{ccc}
\mathbf{u} & t_i' & \mathbf{v}' \\
\mathbf{u}' & t_i' & \mathbf{w}' \\
\mathbf{u}' & t_i & \mathbf{w} \\
\hline
\mathbf{u} & t_i & \varphi(\mathbf{v}', \mathbf{w}', \mathbf{w})\,.
\end{array}$$

Therefore, $(t_1, \dots, t_i) \in \mathsf{pr}_{[i]} \mathsf{cl}_\varphi F$, continuing the induction. □

Given φ and the matrix for F, the procedure of Lemma 15 can be used to decide $\mathbf{t} \in R$ in time $\mathcal{O}(n^2)$. There is no need to generate the whole of R; we just keep track of the tuple $(\mathbf{u}, t_i, \varphi(\mathbf{v}', \mathbf{w}', \mathbf{w}))$ that witnesses that $(t_1, \dots, t_i) \in \mathsf{pr}_{[i]} \mathsf{cl}_\varphi F$. If the procedure succeeds, we have demonstrated that $\mathbf{t} \in \mathsf{cl}_\varphi F = R$; otherwise, we conclude either that $t \notin R$ or that R is not strongly rectangular.

We now show how, given a frame for R, we can determine a frame for the relation

$$R(a_1, \dots, a_i, x_{i+1}, \dots, x_n) = \{\mathbf{t} \in R : (t_1, \dots, t_i) = (a_1, \dots, a_i)\}\,.$$

Lemma 16. *Given a small frame F for $R(x_1, x_2, \dots, x_n)$, we can construct a frame for $R(a, x_2, \dots, x_n)$ in $\mathcal{O}(n^2)$ time.*

Proof. We abbreviate $R(a, x_2, \dots, x_n)$ to $R(a, \cdot)$. For each $i = 2, \dots, n$, determine $\mathsf{cl}_\varphi \mathsf{pr}_{1,i} F = \mathsf{pr}_{1,i} \mathsf{cl}_\varphi F = \mathsf{pr}_{1,i} R$. Note that $|\mathsf{pr}_{1,i} R| \leq q^2$ and $\|F\| = \mathcal{O}(n)$ so this requires $\mathcal{O}(n)$ time for each i, and $\mathcal{O}(n^2)$ time in total. We have $(a, b) \in \mathsf{pr}_{1,i} R$ if, and only if, $b \in \mathsf{pr}_i R(a, \cdot)$. Also, we have calculated a witness (with respect to R) for each $b \in \mathsf{pr}_i R(a, \cdot)$. Let $\sim_i$ be the usual congruence for R and $\sim_i'$ the corresponding congruence for $R(a, \cdot)$. Clearly $b \sim_i' c$ implies $b \sim_i c$, since there are witnesses

$(a, \mathbf{u}, b, \mathbf{v}), (a, \mathbf{u}, c, \mathbf{v}') \in R$. On the other hand, if $b \sim_i c$ and $b \in \mathsf{pr}_i R(a, \cdot)$, then $c \in \mathsf{pr}_i R(a, \cdot)$ and $b \sim_i' c$, since we have

$$\begin{array}{cccc} a & \mathbf{u} & b & \mathbf{v} \\ a' & \mathbf{u}' & b & \mathbf{v}' \\ a' & \mathbf{u}' & c & \mathbf{v}'' \\ \hline a & \mathbf{u} & c & \varphi(\mathbf{v}, \mathbf{v}', \mathbf{v}'')\,. \end{array}$$

Thus, the equivalence classes of $\sim_i'$ are a subset of those of $\sim_i$. Therefore we can easily construct $\sim_i'$ and a witness for each $b \in \mathsf{pr}_i R(a, \cdot)$, using F and the n-tuples from the calculation of $\mathsf{pr}_{1,i} R$. □

The following corollary is immediate, by iterating the Lemma 16 $i \leq n$ times.

Corollary 17. *Given a frame F for the relation $R(x_1, x_2, \dots, x_n)$, a frame for $R(a_1, \dots, a_i, x_{i+1}, \dots, x_n)$ can be constructed in $\mathcal{O}(n^3)$ time.*

5. Constructing a frame. If R is Γ-definable, then $\mathbf{t} \in R$ can be decided in polynomial time by checking that $\mathbf{t}$ satisfies each of the defining constraints. We cannot use this method to decide $R = \emptyset$ efficiently but this can be done trivially using any frame F for R, since $R = \emptyset$ if, and only if, $F = \emptyset$. If $F \neq \emptyset$, then any $\mathbf{f} \in F$ is a certificate that $R \neq \emptyset$. Similarly, given a frame for R and any tuple $(a_1, \dots, a_i)$, we can determine whether there is any $\mathbf{t} \in R$ such that $(t_1, \dots, t_i) = (a_1, \dots, a_i)$, using the method of Corollary 17.

However, we must be able to construct some frame F for R efficiently. If Γ is strongly rectangular, we will show how to determine a frame for a Γ-formula Φ having m constraints in n variables, in time polynomial in m, n and $\|\Gamma\|$. This is achieved, as in [4], by adding constraints sequentially.

If the m constraints are $\Theta_1, \Theta_2, \dots, \Theta_m$, let $\Phi_s = \Theta_1 \wedge \Theta_2 \wedge \cdots \wedge \Theta_s$. Thus, $\Phi_0 = D^n$, the complete n-ary relation on D, and $\Phi_m = \Phi$. We begin by constructing a frame for Φ_0.

Lemma 18. *A small frame F_0 for Φ_0 can be constructed in $\mathcal{O}(n)$ time.*

Proof. Let d be any element of D and let $F_0 = \{\mathbf{t}^d\} \cup \{\mathbf{t}^{a,i} : i \in [n], a \in D \setminus d\}$, where

$$t_j^d = d \quad \text{and} \quad t_j^{a,i} = \begin{cases} a & \text{if } j = i \\ d & \text{otherwise} \end{cases} \qquad (j \in [n]).$$

Clearly all these tuples are in Φ_0. Also $\boldsymbol{\omega}(d, i) = \mathbf{t}^d$ and $\boldsymbol{\omega}(a, i) = \mathbf{t}^{a,i}$ $(a \neq d)$, for all $i \in [n]$, is a witness function. Further, we have $\mathsf{pr}_{[i-1]} \mathbf{t}^{a,i} = \mathsf{pr}_{[i-1]} \mathbf{t}^d = (d, \dots, d)$. Thus, F_0 satisfies the conditions for being a frame. We have $|F_0| = n(q-1) + 1$, so F_0 is small. □

Note that $|F_0|$ matches the upper bound for the size of a small frame.

Now, we show how to determine a frame for Φ_s given a frame for Φ_{s-1}. We first show that this can be done in polynomial time when $\|\Gamma\| = \mathcal{O}(1)$. This is nonuniform CSP, the most important case.

Lemma 19. *Given a frame F for Φ and a constraint Θ, a frame F' for $\Phi' = \Phi \wedge \Theta$ can be constructed in $\mathcal{O}(n^4)$ time.*

Proof. Suppose that $\Theta = H(x_{i_1}, x_{i_2}, \dots, x_{i_r})$, where $H \in \Gamma$ has arity r. We will assume that $x_{i_1}, x_{i_2}, \dots, x_{i_r}$ are distinct since, otherwise, we can consider a smaller relation H' over the distinct variables. Let $I = \{i_1, i_2, \dots, i_r\}$. For each $i \in [n]$, let $J_i = I \cup \{i\}$ and determine $T_i \subseteq \Phi$ such that $\mathsf{pr}_{J_i} T_i = \mathsf{cl}_\varphi \mathsf{pr}_{J_i} \Phi$ using Closure. If $\ell = |\mathsf{pr}_I \Phi|$, then $|T_i| \leq q\ell$, so this takes time $\mathcal{O}(n\ell^3 + r\ell^4)$ by Lemma 13. But, since

$\|\Gamma\| = \mathcal{O}(1)$, we have $r = \mathcal{O}(1)$, $\ell \le q^r = \mathcal{O}(1)$ and $\mathcal{O}(n\ell^3 + r\ell^4) = \mathcal{O}(n)$. The entire computation for all i therefore takes time $\mathcal{O}(n^2)$ and we have $\sum_i |T_i| = \mathcal{O}(n)$.

Determine U_i, the set of tuples in T_i that are consistent with Θ, so $U_i \subseteq \Phi'$. Now U_i contains a witness for each $a \in \mathsf{pr}_i\Phi'$, since

$$\mathsf{pr}_{J_i}U_i = (\mathsf{pr}_{J_i}T_i) \cap \Theta = (\mathsf{cl}_\varphi \mathsf{pr}_{J_i}F) \cap \Theta = (\mathsf{pr}_{J_i}\Phi) \cap \Theta = \mathsf{pr}_{J_i}(\Phi \wedge \Theta) = \mathsf{pr}_{J_i}\Phi' \,.$$

Thus, in particular, $\mathsf{pr}_iU_i = \mathsf{pr}_i\Phi'$. We now do the following for each $i \in [n]$.

Let $\mathcal{A} \leftarrow \mathsf{pr}_iU_i$ and repeat the following until $\mathcal{A} = \emptyset$. Choose $\mathbf{t} \in U_i$ such that $t_i \in \mathcal{A}$. Determine a frame $F^\star$ for $\Phi(t_1, \ldots, t_{i-1}, x_i, \ldots, x_n)$ in $\mathcal{O}(n^3)$ time, using Corollary 17. Clearly $\mathbf{t} \in \mathsf{cl}_\varphi F^\star$, so $F^\star \neq \emptyset$. Now determine the intersection of Θ with the relation $R^\star = \Phi(t_1, \ldots, t_{i-1}, x_i, \ldots, x_n)$ generated by $F^\star$, using Closure, as was done for Φ above. This takes $\mathcal{O}(n)$ time; let the resulting relation be R°. Now, by Corollary 11, pr_iR° is the equivalence class $\mathcal{E} = \{a : a \sim_i' t_i\}$ of t_i in Φ'. For each $a \in \mathcal{E}$, we can find a witness $\boldsymbol{\omega}'(a, i) \in R^\circ$ for $a \in \mathsf{pr}_i\Phi'$ and these have the common prefix $(t_1, \ldots, t_{i-1})$. We set $\mathcal{A} \leftarrow \mathcal{A} \setminus \mathcal{E}$, and repeat.

At the end of this process, $\boldsymbol{\omega}'$ is the witness function for a frame F' for Φ'. The total time required is $\mathcal{O}(n^3|F'|) = \mathcal{O}(n^4)$. □

Lemma 20. *A frame F for Φ can be constructed in time $\mathcal{O}(mn^4)$.*

Proof. Construct Φ_0 in $\mathcal{O}(n)$ time. Then, apply Lemma 19 to construct a frame F_i for Φ_i from a frame F_{i-1} for Φ_{i-1}, for each $i \in [m]$. At termination, set $\Phi \leftarrow \Phi_m$ and $F \leftarrow F_m$. □

Since a relation has $\emptyset$ for a frame if, and only if, it is empty (and $\emptyset$ has no other frame), we can determine in time $\mathcal{O}(mn^4)$ whether there is a satisfying assignment to a CSP instance in a fixed strongly rectangular vocabulary. By Lemma 5, we have re-proven the main result of [4].

We assumed above that $\|\Gamma\| = \mathcal{O}(1)$. However, we can still perform the computations of Lemma 19 in time polynomial in m, n and $\|\Gamma\|$.

Lemma 21. *A frame for Φ can be constructed in time $\mathcal{O}(mn^4 + mn^2\|\Gamma\|^4)$.*

Proof. We indicate how the proof of Lemma 19 must be modified. It is only the computation of the U_i that requires improvement, which we achieve by using a device from [4]. Suppose we wish to add a constraint $\Theta = H(x_{i_1}, x_{i_2}, \ldots, x_{i_r})$ to Φ. Instead, we add in turn the r constraints $\Theta_k = H_k(x_{i_1}, x_{i_2}, \ldots, x_{i_k})$, where $H_k = \mathsf{pr}_{[k]}H$ for each $k \in [r]$. Thus, $|H_1| \le q$ and $H_r = H$. Letting $\Psi_0 = \Phi$, we successively calculate frames for $\Psi_k = \Psi_{k-1} \wedge \Theta_k$ $(k \in [r])$, so $\Psi_r = \Phi'$.

If $I_k = \{i_1, i_2, \ldots, i_k\}$ $(k \in [r])$, we have

$$\ell_k \;=\; |\mathsf{pr}_{I_k}\Psi_{k-1}| \;\le\; q|\mathsf{pr}_{I_{k-1}}\Psi_{k-1}| \;\le\; q|H_{k-1}| \;\le\; q|H| \,.$$

Thus, for each $k \in [r]$, the time required to compute U_i and R° in Lemma 19 becomes $\mathcal{O}(n^2|H|^3 + nr|H|^4)$. In total, the time requirement is $\mathcal{O}(n^2r|H|^3 + nr^2|H|^4) = \mathcal{O}(n^2\|H\|^4) = \mathcal{O}(n^2\|\Gamma\|^4)$. □

6. Counting problems. We consider the problem of determining $|R_\Phi|$, which we abbreviate to $|\Phi|$, where Φ is a Γ-formula with m constraints and n variables. We require the computations to be done in time polynomial in the size of the input Φ and we assume $\|\Gamma\| = \mathcal{O}(1)$. In fact, the size of Φ can be measured by a polynomial in n. A repeat of a constraint can be removed, since this does not change R_Φ. Then an r-ary relation in Γ can give rise to $\mathcal{O}(n^r)$ constraints. We will assume that every variable appears in at least one constraint. Otherwise, suppose n_0 variables do not appear: letting Φ' be Φ with these variables deleted, we have $|\Phi| = q^{n_0}|\Phi'|$. Hence we will assume that $m = \Omega(n)$.

Following Bulatov and Dalmau [6], we call this computational problem $\#\mathsf{CSP}(\Gamma)$. If $\Gamma = \{H, =\}$, we write $\#\mathsf{CSP}(H)$. We will use the following result from [6], which we prove here for completeness. The corollary is immediate.

THEOREM 22 (Bulatov and Dalmau [6]). *Let $\mathfrak{S} = (D, \Gamma)$, $\mathfrak{S}' = (D, \Gamma')$ be relational structures with $\Gamma' \subseteq \langle\Gamma\rangle$. Then $\#\mathsf{CSP}(\Gamma')$ is polynomial-time reducible to $\#\mathsf{CSP}(\Gamma)$.*

Proof. Let each $H' \in \Gamma'$ have pp-definition $H'(\mathbf{x}) = \exists \mathbf{y}\, H^*(\mathbf{x}, \mathbf{y})$, with $H^*(\mathbf{x}, \mathbf{y})$ a Γ-formula. If all relations in Γ have arity at most r and at most ℓ tuples and all the formulae H^* are conjunctions of at most k constraints, then each H^* has arity at most kr and $|H^*| \leq \ell^k$. Observe that k, ℓ and r are constants in $\#\mathsf{CSP}(\Gamma')$.

Consider any Γ' formula $\Phi(\mathbf{x}) = \Theta_1 \wedge \cdots \wedge \Theta_m$, where $\mathbf{x} = (x_1, \ldots, x_n)$. Now, if $\Theta_i = H'(\mathbf{x})$, let $\Theta_i^* = H^*(\mathbf{x}, \mathbf{y}_i)$, where the $\mathbf{y}_i$ ($i \in [m]$) are new variables. Let $\mathbf{z} = (\mathbf{y}_1, \ldots, \mathbf{y}_m)$ and consider the Γ-formula $\Phi^*(\mathbf{x}, \mathbf{z}) = \Theta_1^* \wedge \cdots \wedge \Theta_m^*$. This is an instance of $\#\mathsf{CSP}(\Gamma)$, with at most km constraints and $n + krm$ variables. Now, for $\mathbf{x} \in \Phi$, let

$$N_i(\mathbf{x}) \ = \ \big|\{\mathbf{y}_i : (\mathbf{x}, \mathbf{y}_i) \in \Theta_i^*\}\big| \ \leq \ |H^*| \ \leq \ell^k \qquad (i \in [m]),$$

and let $N = \max\{N_i(\mathbf{x}) : i \in [m],\, \mathbf{x} \in \Phi\} \leq \ell^k$. Now let

$$\mu_j(\mathbf{x}) \ = \ \big|\{i \in [m] : N_i(\mathbf{x}) = j\}\big| \qquad (j \in [N]).$$

Clearly $\sum_{j=1}^N \mu_j(\mathbf{x}) = m$ for all $\mathbf{x} \in \Phi$. Let

$$\mathbf{M} \ = \ \{(\mu_1(\mathbf{x}), \ldots, \mu_N(\mathbf{x})) : \mathbf{x} \in \Phi\}\,.$$

Let $L = |\mathbf{M}|$. Clearly, $|\mathbf{M}| < m^N$, so L has bit-size $\mathcal{O}(m)$. Now, for $\mathbf{m} \in \mathbf{M}$, let

$$K(\mathbf{m}) \ = \ \big|\{\mathbf{x} \in \Phi : \mu_j(\mathbf{x}) = m_j,\, j \in [N]\}\big| \ \leq \ q^n \ \leq \ q^m\,.$$

Thus, $|\Phi| = \sum_{\mathbf{m}\in\mathbf{M}} K(\mathbf{m})$. Now let $J(\mathbf{m}) = \prod_{j=1}^N j^{m_j} < N^m$. Thus, the $J(\mathbf{m})$, $K(\mathbf{m})$ ($\mathbf{m} \in [\mathbf{M}]$) are numbers with $\mathcal{O}(m)$ bits. Then we have

$$|\Phi^*| \ = \ \sum_{\mathbf{x}\in\Phi} \prod_{i\in[m]} N_i(\mathbf{x}) \ = \ \sum_{\mathbf{m}\in\mathbf{M}} K(\mathbf{m}) \prod_{j=1}^{N} j^{m_j} \ = \ \sum_{\mathbf{m}\in\mathbf{M}} K(\mathbf{m}) J(\mathbf{m})\,.$$

Now, for $s \in [L]$, consider the Γ-formulae

$$\Phi_s^*(\mathbf{x}, \mathbf{z}_1, \ldots, \mathbf{z}_s) \ = \ \bigwedge_{i\in[s]} \Phi^*(\mathbf{x}, \mathbf{z}_i)\,,$$

where $\mathbf{z}_i$ ($i \in [s]$) are distinct variables. Then Φ_s^* is an instance of $\#\mathsf{CSP}(\Gamma)$, with at most kms constraints and $krms$ variables, and we clearly have

$$|\Phi_s^*| \ = \ \sum_{\mathbf{m}\in\mathbf{M}} K(\mathbf{m}) J(\mathbf{m})^s.$$

Note that Φ_s^* is of size polynomial in m. Therefore we can evaluate $|\Phi_s^*|$ for all $s \in [L]$ using a polynomial number of calls to an oracle for $\#\mathsf{CSP}(\Gamma)$, each having input of size polynomial in m. It then follows, using [16, Lemma 3.2], that we can recover $\sum_{\mathbf{m}\in\mathbf{M}} K(\mathbf{m}) = |\Phi|$ from the values of the $|\Phi_s^*|$ ($s \in [L]$) in time polynomial in L, which is polynomial in m. $\square$

Corollary 23. *If $H \in \langle \Gamma \rangle$ and $\#\mathsf{CSP}(H)$ is $\#\mathsf{P}$-complete, then $\#\mathsf{CSP}(\Gamma)$ is $\#\mathsf{P}$-complete.* □

First, we apply Corollary 23 to give a short proof of the main result of [6]. (Bulatov and Dalmau phrase the result in terms of the existence of a Mal'tsev polymorphism but, by Lemma 5, our phrasing is equivalent.)

Lemma 24 (Bulatov and Dalmau [6]). *If the constraint language Γ is not strongly rectangular, then $\#\mathsf{CSP}(\Gamma)$ is $\#\mathsf{P}$-complete.*

Proof. Clearly $\#\mathsf{CSP}(\Gamma) \in \#\mathsf{P}$ for any Γ. If Γ is not strongly rectangular, there is an r-ary relation $B \in \langle \Gamma \rangle$ that is not rectangular when considered as a binary relation over $D^k \times D^{r-k}$ for some k with $1 \leq k < r$. Let $G = (V, E)$ be a connected, undirected bipartite graph with vertex bipartition V_1, V_2. Let Φ_1 be the Γ-formula with a constraint $B(\mathbf{x}_i, \mathbf{x}_j)$ for each $\{\nu_i, \nu_j\} \in E$ with $\nu_i \in V_1$, $\nu_j \in V_2$. Define Φ_2 analogously, but with constraints $B(\mathbf{x}_j, \mathbf{x}_i)$. It follows that $|\Phi_1| + |\Phi_2|$ is the number of graph homomorphisms from G to $\mathcal{G}_B$. This problem is $\#\mathsf{P}$-complete by [16], since $\mathcal{G}_B$ has a component which is not a bipartite clique. Thus, $\#\mathsf{CSP}(B)$ is $\#\mathsf{P}$-complete and, hence, $\#\mathsf{CSP}(\Gamma)$ is $\#\mathsf{P}$-complete by Corollary 23. □

There is an important generalisation of the counting problem to *weighted* problems which we now describe briefly; see [8,14] for details. The relations $H \subseteq D^r$ in Γ are replaced by functions $f \colon D^r \to \mathbb{Q}^+$, where $\mathbb{Q}^+$ denotes the non-negative rationals.[2] Thus, Γ is replaced by a set of functions $\mathcal{F}$. We will call $(D, \mathcal{F})$ a *weighted structure.* The *underlying relation* of $f \in \mathcal{F}$ is $\{\mathbf{u} \in D^r : f(\mathbf{u}) > 0\}$. Note that a relation H can be identified with a function $f_H \colon D^r \to \{0, 1\}$, where $f_H(\mathbf{u}) = 1$ if, and only if, $\mathbf{u} \in H$. Then H is the underlying relation of f_H. Thus, we may just use H to denote the function f_H without further comment.

Now, using notation similar to the relational case, an instance $\mathcal{I}$ of $\#\mathsf{CSP}(\mathcal{F})$ is defined as follows. A constraint Θ has the form $f(x_{i_1}, x_{i_2}, \ldots, x_{i_r})$ for some r-ary function $f \in \mathcal{F}$. Thus, $(\nu_{i_1}, \nu_{i_2}, \ldots, \nu_{i_r})$ is the scope of the Θ. Suppose we have constraints $\Theta_1, \ldots, \Theta_m$, where Θ_s applies the function $f_s \in \mathcal{F}$. Write $\mathbf{x}_s$ for $(x_{i_1}, x_{i_2}, \ldots, x_{i_r})$, where $(\nu_{i_1}, \nu_{i_2}, \ldots, \nu_{i_r})$ is the scope of the Θ_s. Then, the *weight* of an assignment $\mathbf{x} \colon V \to D$ is

$$\mathsf{W}(\mathbf{x}) = \prod_{s=1}^{m} f_s(\mathbf{x}_s)\,.$$

The computational problem $\#\mathsf{CSP}(\mathcal{F})$ is then to compute the *partition function*,

$$Z(\mathcal{I}) = \sum_{\mathbf{x} \colon V \to D} \mathsf{W}(\mathbf{x})\,.$$

If $\mathcal{F} = \{f\}$ for a single function f, we write $\#\mathsf{CSP}(f)$.

We may view a binary function $f \colon A_1 \times A_2 \to \mathbb{Q}^+$ as a matrix with elements in $\mathbb{Q}^+$, rows indexed by A_1 and columns indexed by A_2. If B is its underlying relation, the submatrix of f induced by a block of B is called a block of f. If $f_1, f_2, \ldots, f_k$ are the blocks of f, then f will be called a *rank-one block matrix,* if each block of f is a rank one matrix.

Lemma 25. *If $f \colon A_1 \times A_2 \to \mathbb{Q}^+$ is a rank-one block matrix, its underlying relation B is rectangular.*

Proof. If B is not rectangular, there are (a, c), (b, c), $(a, d) \in R$ such that $(b, d) \notin B$. The 2×2 sub-matrix of f induced by rows a, b and columns c, d is included within

[2] More generally, we can take the function values to be non-negative algebraic numbers.

a single block and has determinant $-f(a,d)f(b,c) \neq 0$ and so has rank 2. Therefore, f has a block of rank at least 2. □

We will call a matrix $f\colon A_1 \times A_2 \to \mathbb{Q}^+$ *rectangular* if its underlying relation R is rectangular. Thus, an alternative way of defining a rank-one block matrix is as a rectangular matrix f, together with functions $\alpha_1\colon A_1 \to \mathbb{Q}^+$, $\alpha_2\colon A_2 \to \mathbb{Q}^+$, such that $f(x,y) = \alpha_1(x)\alpha_2(y)$ for all $(x,y) \in B$.

We can now state a theorem of Bulatov and Grohe [8, Theorem 14], which generalises the result of Dyer and Greenhill [16] to the weighted case. Although we give the theorem for non-negative rational functions, in fact we only require the case for non-negative integer functions.

THEOREM 26 (Bulatov and Grohe [8]). *Let $f\colon A_1 \times A_2 \to \mathbb{Q}^+$ be a binary function. Then $\#\mathsf{CSP}(f)$ is in FP if f is a rank-one block matrix. Otherwise $\#\mathsf{CSP}(f)$ is $\#\mathsf{P}$-hard.* □

In Section 7.1, we will use the following property of rank-one block matrices.

LEMMA 27. *If $f\colon A_1 \times A_2 \to \mathbb{Q}^+$ is a rank-one block matrix, it is uniquely determined by its underlying relation and its row and column totals.*

Proof. Let B be the underlying (rectangular) relation. Consider any block C of B, with $\mathsf{pr}_1 C = S_1$, $\mathsf{pr}_2 C = S_2$. Then there exist $\alpha_1\colon S_1 \to \mathbb{Q}^+$ and $\alpha_2\colon S_2 \to \mathbb{Q}^+$ such that $f(x_1,x_2) = \alpha_1(x_1)\alpha_2(x_2)$ for every $x_1 \in S_1$ and $x_2 \in S_2$. Now, let

$$\begin{aligned}
f(x_1,\cdot) &= \sum_{x_2 \in S_2} f(x_1,x_2) &&= \alpha_1(x_1) \sum_{x_2 \in S_2} \alpha_2(x_2)\\
f(\cdot,x_2) &= \sum_{x_1 \in S_1} f(x_1,x_2) &&= \alpha_2(x_2) \sum_{x_1 \in S_1} \alpha_1(x_1)\\
f(\cdot,\cdot) &= \sum_{x_1 \in S_1} f(x_1,\cdot) &&= \sum_{x_1 \in S_1} \alpha_1(x_1) \sum_{x_2 \in S_2} \alpha_2(x_2)
\end{aligned}$$

be the row, column and grand totals of $f(x_1,x_2)$ $(x_1 \in S_1, x_2 \in S_2)$. A simple calculation gives

$$f(x_1,x_2) = \frac{f(x_1,\cdot)f(\cdot,x_2)}{f(\cdot,\cdot)}. \qquad □$$

7. The dichotomy theorem. We are now ready to describe the dichotomy. We saw in the previous section that, assuming $\mathsf{FP} \neq \#\mathsf{P}$, strong rectangularity is a necessary condition for tractability. In this section, we introduce a stronger condition, based on certain rank-one block matrices and show that it characterises the dichotomy for #CSP, into problems in FP and problems which are #P-complete. As one would expect, this condition turns out to be equivalent to the criterion in Bulatov's dichotomy theorem. We defer the algorithm for the polynomial-time cases to Section 7.1 and some technical results to Section 7.2. In Section 8, we will show that the condition is decidable.

Let $H(x,y,z)$ be a ternary relation on $A_1 \times A_2 \times A_3$. We will call H *balanced* if the *balance matrix*

$$M(x,y) = |\{z \in A_3 : (x,y,z) \in H\}| \qquad (x \in A_1,\ y \in A_2)$$

is a rank-one block matrix. A relation of arity $n > 3$ is balanced if every expression of it as a ternary relation on $D^k \times D^\ell \times D^{n-k-\ell}$ $(d, \ell \geq 1,\ k + \ell < n)$ is balanced. We will say that Γ is *strongly balanced* if every pp-definable ternary relation is balanced.

We will prove the following dichotomy theorem.

THEOREM 28. *If* Γ *is strongly balanced,* $\#\mathsf{CSP}(\Gamma)$ *is in* FP. *Otherwise,* $\#\mathsf{CSP}(\Gamma)$ *is* $\#\mathsf{P}$*-complete. Moreover, the dichotomy is decidable.*

Proof. The first statement will be proved in Section 7.1. The second is proved in Lemma 31 below. The third is proved in Section 8. □

We first show that the condition of strong balance is strictly stronger than that of strong rectangularity.

LEMMA 29. *Strong balance implies strong rectangularity.*

Proof. This follows from the definition of strong balance. Suppose Γ is strongly balanced and let $B(x, y)$ be any definable binary relation. Let

$$H(x, y, z) = \exists w\, B(x, y) \wedge B(z, w)\,,$$

which must be balanced. Then $M(x, y) = |\{z : \exists w\, B(z, w)\}| = |\mathsf{pr}_1 B|$, for all $(x, y) \in B$. If $|\mathsf{pr}_1 B| = 0$ then $B = \emptyset$, which is trivially rectangular. Otherwise, the underlying relation of M is B, which must be rectangular by Lemma 25. □

The converse of Lemma 29 is not true, however.

LEMMA 30. *Strong rectangularity does not imply strong balance.*

Proof. Consider the following example. Let $A = \{a_{0,0}, a_{0,1}, a_{1,0}, a_{1,1}, b\}$ and let $D = A \cup \{0, 1\}$. Let $\Gamma = \{R\}$, where R is the ternary relation given by

$$R = \{(i, j, a_{i,j}) : i, j \in \{0, 1\}\} \cup \{(0, 0, b)\}\,.$$

Note that b is, in effect, a second copy of $a_{0,0}$; the effect is essentially that of a weighted relation where the tuple $(0, 0, a_{0,0})$ has weight 2 and all other tuples have unit weight. The balance matrix M for R is as follows (we omit the rows and columns for $x \in A$ as they have only zeroes):

$$M \;=\; \begin{array}{c} \\ 0 \\ 1 \end{array}\!\!\begin{array}{c} \begin{array}{cc} 0 & 1 \end{array} \\ \left[\begin{array}{cc} 2 & 1 \\ 1 & 1 \end{array}\right] \end{array}.$$

M is clearly not a rank-1 block matrix, so R is not strongly balanced. Nonetheless, we will show that R has a Mal'tsev polymorphism. Consider the following function, where $\oplus$ denotes addition modulo 2.

$$f(x, y, z) = \begin{cases} x \oplus y \oplus z & \text{if } x, y, z \in \{0, 1\} \\ a_{f(i,k,m), f(j,\ell,n)} & \text{if } x = a_{i,j}, y = a_{k,\ell}, z = a_{m,n} \\ a_{0,0} & \text{otherwise.} \end{cases}$$

Let $g(b) = a_{0,0}$ and $g(x) = x$ for all other $x \in D$. We define the function φ as follows:

$$\varphi(x, y, z) = \begin{cases} x & \text{if } y = z \\ z & \text{if } x = y \\ f(g(x), g(y), g(z)) & \text{otherwise.} \end{cases}$$

In other words, φ behaves identically to f, except that it has the Mal'tsev property and, for inputs where $x \neq y$ and $y \neq z$, it "pretends" that any input of b is actually an input of $a_{0,0}$. Note that, for $i, j, k \in \{0, 1\}$, $\varphi(i, j, k) = i \oplus j \oplus k$, regardless of the Mal'tsev condition.

We claim that, as well as being Mal'tsev, φ is a polymorphism of R. To this end, let $\mathbf{x}, \mathbf{y}, \mathbf{z} \in R$, which we can write as $\mathbf{x} = (i, j, x')$, $\mathbf{y} = (k, \ell, y')$ and $\mathbf{z} = (m, n, z')$, where $x' = a_{i,j}$ or, if $i = j = 0$, we may have $x' = b$, and similarly for y' and z'. So, we have

$$\begin{aligned}\varphi(\mathbf{x}, \mathbf{y}, \mathbf{z}) &= \big(\varphi(i,k,m), \varphi(j,\ell,n), \varphi(x',y',z')\big)\\ &= \big(f(i,k,m), f(j,\ell,n), f(g(x'), g(y'), g(z')\big)\\ &= \big(f(i,k,m), f(j,\ell,n), a_{f(i,k,m),f(j,\ell,n)}\big)\\ &\in R\,.\end{aligned}$$

This establishes the claim. □

Remark 7. The example in Lemma 30 can be extended to relations of arbitrary size by extending i and j in the tuples $(i, j, a_{i,j})$ to longer binary strings and interpreting $\oplus$ as bit-wise XOR (e.g., $0011 \oplus 0101 = 0110$).

Remark 8. Bulatov and Dalmau conjectured in [5] that a Mal'tsev polymorphism was sufficient for $\#\mathsf{CSP}(\Gamma)$ to be in FP. That is a stronger claim than the converse of Lemma 29. The conjecture was withdrawn in [6], with a counterexample somewhat similar to that in the proof of Lemma 30.

Next, we strengthen Lemma 24 to prove one half of the dichotomy.

Lemma 31. *If Γ is not strongly balanced, then $\#\mathsf{CSP}(\Gamma)$ is $\#\mathsf{P}$-complete.*

Proof. If Γ is not strongly balanced, there is an unbalanced ternary relation $H \in \langle\Gamma\rangle$. Let E be a binary relation with $V = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$ and $\mathsf{pr}_i E = V_i$ $(i = 1, 2)$. Let Φ be the Γ-formula with a constraint $H(x_i, x_j, z_{ij})$ for each $(\nu_i, \nu_j) \in E$. Thus, Φ has $|V| + |E|$ variables and $|E|$ constraints. Let $M \colon V_1 \times V_2 \to \mathbb{Q}^+$ be Φ's balance matrix.

We have $|\Phi| = Z(\mathcal{I})$, where $Z(\mathcal{I})$ is the partition function for an instance $\mathcal{I}$ of $\#\mathsf{CSP}(M)$ with input E. But this problem is $\#\mathsf{P}$-hard by Theorem 26 and, hence, $\#\mathsf{CSP}(H)$ is $\#\mathsf{P}$-complete. Thus, $\#\mathsf{CSP}(\Gamma)$ is $\#\mathsf{P}$-complete by Corollary 23. □

In [3], Bulatov defined *congruence singularity.* Suppose Γ is a constraint language and ρ_1 and ρ_2 are two congruences defined on the same pp-definable set $A \subseteq D^r$. Let the equivalence classes of ρ_i be E_{ij} $(j \in [\nu_i],\ i = 1, 2)$. Further, let

$$\mathcal{M}(j,k) = |E_{1j} \cap E_{2k}| \qquad (j \in [\nu_1],\ k \in [\nu_2]). \tag{1}$$

Then Γ is *congruence singular* if $\mathcal{M}$ is a rank-one block matrix for every pair ρ_1, ρ_2 of congruences.[3]

Lemma 32. *Γ is congruence singular if, and only if, it is strongly balanced.*

Proof. Suppose Γ is strongly balanced, let $A \subseteq D^r$ be defined by the formula χ and let $\rho_1, \rho_2 \in \langle\Gamma\rangle$ be congruences defined on $A \subseteq D^r$ with equivalence classes E_{ij} $(j \in [\nu_i],\ i = 1, 2)$. Then $\psi(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \chi(\mathbf{z}) \wedge \rho_1(\mathbf{x}, \mathbf{z}) \wedge \rho_2(\mathbf{z}, \mathbf{y})$ is a ternary relation. Hence, for any $\mathbf{x} \in E_{1j}$ and $\mathbf{y} \in E_{2k}$, the matrix

$$M(\mathbf{x}, \mathbf{y}) = |\{\mathbf{z} : \chi(\mathbf{z}) \wedge \rho_1(\mathbf{x}, \mathbf{z}) \wedge \rho_2(\mathbf{z}, \mathbf{y})\}| = |E_{1j} \cap E_{2k}|$$

is a rank-one block matrix. But M has a set of identical rows for all $\mathbf{x} \in E_{1j}$ $(j \in [\nu_1])$ and a set of identical columns for all $\mathbf{y} \in E_{2k}$ $(k \in [\nu_2])$. The matrix $\mathcal{M}$ has one representative from each of these sets. It follows that $\mathcal{M}$ is a rank-one block matrix.

Now, suppose that Γ is congruence singular and let $H \in \langle\Gamma\rangle$ be any ternary relation. Define relations $\rho_i = \{(\mathbf{x}, \mathbf{y}) : \mathbf{x}, \mathbf{y} \in H \text{ and } x_i = y_i\}$ $(i = 1, 2)$. These are

[3] In fact, Bulatov applies this term to the associated algebra, but with essentially this meaning.

trivially equivalence relations, and are pp-definable as $H(x_1, x_2, x_3) \wedge H(y_1, y_2, y_3) \wedge (x_i = y_i)$. Thus, they are two congruences defined on the same set, H, which is also pp-definable. The equivalence classes of ρ_i clearly correspond to $z_i \in \mathsf{pr}_i H$ $(i = 1, 2)$ and we may index these classes by z_i. Thus,

$$\begin{aligned}\mathcal{M}(z_1, z_2) &= |\{(x_1, x_2, x_3) \in H : x_1 = z_1, x_2 = z_2\}| \\ &= |\{x_3 : (z_1, z_2, x_3) \in H\}| \\ &= M(z_1, z_2)\,.\end{aligned}$$

Since $\mathcal{M}$ is a rank-one block matrix by assumption, so is M, and the conclusion follows. □

In [3], Bulatov established the following theorem, giving a dichotomy for #CSP that is equivalent, using Lemma 32, to Theorem 28, except that the decidability of the dichotomy remained open.

THEOREM 33 (Bulatov [3]). *If Γ is congruence singular, $\#\mathsf{CSP}(\Gamma)$ is in* FP. *Otherwise $\#\mathsf{CSP}(\Gamma)$ is #P-complete.*

7.1. The counting algorithm. This section is devoted to a proof of the polynomial-time case of the dichotomy theorem.

LEMMA 34. *Let Γ be strongly balanced and let $R \in \langle\Gamma\rangle$ be an n-ary relation. Given a frame F for R, $|R|$ can be computed in $\mathcal{O}(n^5)$ time.*

Proof. If $n = 1$ then $R = \mathsf{pr}_1 R = \mathsf{pr}_1 F = F$ so $|R| = |F|$ and we are done. So we may assume that $n \geq 2$. Now, for $1 \leq i < j \leq n$, define $N_{i,j} \colon \mathsf{pr}_j R \to \mathbb{N}$ by

$$N_{i,j}(a) = |\{(\mathbf{u}, a) \in \mathsf{pr}_{[i]\cup\{j\}} R\}|\,.$$

Since we have

$$|R| = \sum_{a \in \mathsf{pr}_n R} N_{n-1,n}(a)\,,$$

we need to compute the function $N_{n-1,n}$, which we do iteratively. For each $j \in \{2, \ldots, n\}$, $N_{1,j}(a) = |\{b \in \mathsf{pr}_1 R : (b, a) \in \mathsf{pr}_{1,j} R\}|$. By Lemma 13, these quantities can be computed by using F to determine $\mathsf{pr}_{1,j} R$, in total time $\mathcal{O}(n^2)$. (Note, in particular, that $|\mathsf{pr}_{1,j} R| \leq q^2 = \mathcal{O}(1)$ and F may be assumed to be small so $|F| \leq \mathcal{O}(n)$.) To continue the iteration, we use $N_{i,i+1}$ and $N_{i,j}$ to compute $N_{i+1,j}$ for $j = i+2, \ldots, n$. We repeat these computations for each $i = 1, \ldots, n-1$.

Consider a particular i and j and suppose that we have computed $N_{i-1,k}$ for all $k \geq i$. Let $J = [i] \cup \{j\}$ and let $H = \mathsf{pr}_J R$, which we will express as a ternary relation

$$H = \{(\mathbf{u}, x, y) \in \mathsf{pr}_J R : \mathbf{u} \in \mathsf{pr}_{[i-1]} R,\ x \in \mathsf{pr}_i R,\ y \in \mathsf{pr}_j R\}\,.$$

Since R is strongly balanced, the matrix

$$M(x, y) = |\{\mathbf{u} \in \mathsf{pr}_{[i-1]} R : (\mathbf{u}, x, y) \in H\}|$$

is a rank-one block matrix. The block structure of M is given by the relation $\mathsf{pr}_{i,j} R$, since if $(x, y) \in \mathsf{pr}_{i,j} R$, there is at least one $\mathbf{t} \in R$ with $\mathsf{pr}_i \mathbf{t} = x$ and $\mathsf{pr}_j \mathbf{t} = y$. By Lemma 13, we can compute $\mathsf{pr}_{i,j} R$ in $\mathcal{O}(n)$ time, using F.

For notational simplicity, let us write $\mathcal{D}_i = \mathsf{pr}_i R$. Consider $M(\cdot, y)$, the y-indexed row of M. We have

$$\sum_{x \in \mathcal{D}_i} M(x, y) = \sum_{x \in \mathcal{D}_i} |\{\mathbf{u} : (\mathbf{u}, x, y) \in H\}| = |\{(\mathbf{u}, x) : (\mathbf{u}, x, y) \in H\}| = N_{i,j}(y)\,. \quad (2)$$

Now observe that the relation $B_y(\mathbf{u},x) = \{(\mathbf{u},x) : (\mathbf{u},x,y) \in H\}$ is rectangular, by Lemma 9. Let us write $S_y(x) = \{\mathbf{u} : (\mathbf{u},x,y) \in H\}$. Then, by Corollary 2, there is an equivalence relation on $\mathcal{D}_j$

$$\theta_y(x_1,x_2) = \exists \mathbf{u} \left(H(\mathbf{u},x_1,y) \wedge H(\mathbf{u},x_2,y) \right)$$

such that $S_y(x_1)$ and $S_y(x_2)$ are equal, if $\theta_y(x_1,x_2)$, and disjoint, otherwise. Thus, if $\mathcal{S}(y) \subseteq \mathcal{D}_i$ contains one representative of each equivalence class of θ_y, then

$$\sum_{x \in \mathcal{S}(y)} M(x,y) = |\{\mathbf{u} : \exists x\, (\mathbf{u},x,y) \in H\}| = N_{i-1,j}(y)\,. \tag{3}$$

Now, suppose that $\theta_y(x_1,x_2)$ and $y' \neq y$. Thus, $H(\mathbf{u},x_1,y)$ and $H(\mathbf{u},x_2,y)$ for some $\mathbf{u}$, so $(x_1,y),(x_2,y) \in C$ for some block C of $\mathsf{pr}_{i,j}R$. There is $\mathbf{u}'$ such that $H(\mathbf{u}',x_1,y')$ if, and only if, $(x_1,y') \in C$. But then we have

$$\begin{array}{ccc} \mathbf{u}' & x_1 & y' \\ \mathbf{u} & x_1 & y \\ \mathbf{u} & x_2 & y \\ \hline \mathbf{u}' & x_2 & y', \end{array}$$

and, hence, $\theta_{y'}(x_1,x_2)$. Thus, the equivalence relations θ_y depend only on the block C containing y. Thus, we may deduce the classes of θ_y from $\mathsf{pr}_{i,j}R$ and those of the relation $\sim_{i,j}$, defined by

$$x_1 \sim_{i,j} x_2 \quad\Longleftrightarrow\quad \exists \mathbf{u}, y \left(H(\mathbf{u},x_1,y) \wedge H(\mathbf{u},x_2,y) \right).$$

We prove in Section 7.2, below, that the $\sim_{i,j}$ are congruences in $\langle R \rangle$. Thus, the matrix M has identical columns corresponding to the equivalence classes of $\sim_{i,j}$.

Similarly, there are identical rows corresponding to the equivalence classes of $\sim_{j,i}$, where

$$y_1 \sim_{j,i} y_2 \quad\Longleftrightarrow\quad \exists \mathbf{u}, x \left(H(\mathbf{u},x,y_1) \wedge H(\mathbf{u},x,y_2) \right).$$

(There is no ambiguity of notation between $\sim_{i,j}$ and $\sim_{j,i}$ since we have $i < j$.)

We prove in Section 7.2 that the $\sim_{j,i}$ are also congruences in $\langle R \rangle$. Now, if $\mathcal{S}'(x)$ contains one representative of each of the classes of the corresponding equivalence relation θ'_x, we have

$$\sum_{y \in \mathcal{S}'(x)} M(x,y) = |\{\mathbf{u} : \exists y\, (\mathbf{u},x,y) \in H\}| = N_{i-1,i}(x)\,. \tag{4}$$

The matrix $\widehat{M}$, obtained by choosing one representative from each of the equivalence classes of $\sim_{i,j}$ and $\sim_{j,i}$, is also a rank-one block matrix. Moreover, we know the block structure, row and column sums of $\widehat{M}$, from $\mathsf{pr}_{i,j}R$, $\sim_{i,j}$, $\sim_{j,i}$, (3) and (4). Hence, by Lemma 27, we can reconstruct all the entries of $\widehat{M}$. Then, using $\mathsf{pr}_{i,j}R$, $\sim_{i,j}$ and $\sim_{j,i}$, we can reconstruct the matrix M. Finally we compute the row sums, as in (2), to give the value of $N_{i,j}(a)$ for each $a \in \mathsf{pr}_j R$.

The time complexity of the algorithm is $\mathcal{O}(n)$ for a given i and j, even in the bit-complexity model. Since there are $\mathcal{O}(n^2)$ pairs i,j, the overall complexity is $\mathcal{O}(n^3)$.

To complete the proof, we must show how to compute the congruences $\sim_{i,j}$ and $\sim_{j,i}$ in $\mathcal{O}(n^5)$ time. We do this in the following section. □

The time complexity of this algorithm is $\mathcal{O}(n^5)$. However, observe that the time needed to compute F is already $\mathcal{O}(mn^4)$. We may assume that $m = \Omega(n)$ as, otherwise, there is a variable, x_1 say, which appears in no constraint. Thus, x_1 can be removed to give a relation $R_1(x_2, \ldots, x_n)$ such that $|R| = q|R_1|$. Therefore, the time complexity of the counting algorithm is no worse than the $\mathcal{O}(mn^4)$ cost of computing the frame F.

7.2. The congruences $\sim_{i,j}$ and $\sim_{j,i}$. We now prove that the relations $\sim_{i,j}$ and $\sim_{j,i}$ used in the proof of Lemma 34 are congruences and that they can be computed efficiently. Let Γ be strongly rectangular and let R be an n-ary relation determined by a Γ-formula Φ. For $1 < i < j \leq n$, recall that

(i) $a \sim_{i,j} b$ $(a, b \in \mathsf{pr}_j R)$ if there are $\mathbf{t}, \mathbf{t}' \in R$ such that $\mathsf{pr}_{[i]}\mathbf{t} = \mathsf{pr}_{[i]}\mathbf{t}'$, $t_j = a$ and $t'_j = b$;

(ii) $a \sim_{j,i} b$ $(a, b \in \mathsf{pr}_i R)$ if there are $\mathbf{t}, \mathbf{t}' \in R$ such that $\mathsf{pr}_J\mathbf{t} = \mathsf{pr}_J\mathbf{t}'$, $t_i = a$ and $t'_i = b$,
where $J = [i-1] \cup \{j\}$.

Lemma 35. *For all $1 < i < j \leq n$, $\sim_{i,j}$ and $\sim_{j,i}$ are congruences in $\langle R \rangle$.*

Proof. Consider the binary relation B defined by $B(\mathbf{u}, y) = \exists \mathbf{z}_1, \mathbf{z}_2\, R(\mathbf{u}, \mathbf{z}_1, y, \mathbf{z}_2)$ on $\mathsf{pr}_{[i]}R \times \mathsf{pr}_j R$. This is rectangular and so induces a congruence θ_2 on $\mathsf{pr}_j R$ by Corollary 3. This congruence is $\sim_{i,j}$.

The proof for $\sim_{j,i}$ is similar, using B defined by $B(\mathbf{u}, y) = \exists \mathbf{z}_1, \mathbf{z}_2\, R(\mathbf{x}, y, \mathbf{z}_1, w, \mathbf{z}_2)$ on $\mathsf{pr}_J R \times \mathsf{pr}_i R$, where $\mathbf{u} = (\mathbf{x}, w)$. □

Lemma 36. *The set of congruences $\sim_{i,j}$ and $\sim_{j,i}$ $(1 < i < j \leq n)$ can be computed in $\mathcal{O}(n^5)$ time.*

Proof. We compute the relations $\sim_{i,j}$, with $i < j$, as follows. From the frame F, we compute $\mathsf{pr}_{i,j}R$. For each $b \in \mathsf{pr}_i R$, this gives a tuple $\mathbf{t}$ such that $\mathsf{pr}_j \mathbf{t} = b$. We now use Corollary 17, to compute a frame $F^\star$ for $R(t_1, \ldots, t_i, x_{i+1}, \ldots, x_n)$ in $\mathcal{O}(n^3)$ time. Now $\mathsf{pr}_j F^\star$ gives the equivalence class of $\sim_{i,j}$ containing b. We repeat this procedure, as in the proof of Lemma 19, until we have determined all the equivalence classes.

There are $\mathcal{O}(n^2)$ pairs i, j with $i < j$ and computing each $\sim_{i,j}$ requires $\mathcal{O}(n^3)$ time. Thus, the we can compute all $\sim_{i,j}$ in $\mathcal{O}(n^5)$ time.

Now consider the relations $\sim_{j,i}$, with $i < j$. For each $a \in \mathsf{pr}_i R$, compute a frame $F_{j,a}$ for the relation $R_{j,a}$ determined by $\Phi \wedge \chi_a(x_j)$. (Recall that χ_a is the relation containing only a and we may assume that $\chi_a \in \Gamma$ by Lemma 9.) From Lemma 19, we can do this in $\mathcal{O}(n^4)$ time, so $\mathcal{O}(n^5)$ time in total. Now, for each $i < j$, determine $\mathsf{pr}_{i,j}R$, using F. This requires $\mathcal{O}(n)$ time for each pair i, j, so $\mathcal{O}(n^3)$ time in total.

Now, for each block C of $\mathsf{pr}_{i,j}R$, choose $a \in \mathsf{pr}_j R$ so that $(x, a) \in C$ for some $x \in \mathsf{pr}_i R$. Then the congruence $\sim_i$ of $R_{j,a}$ gives the equivalence classes of $\sim_{j,i}$ corresponding to C. These can be determined in $\mathcal{O}(n)$ time using $F_{i,a}$. Thus, the total time to compute $\sim_{j,i}$ for all pairs i, j with $i < j$ is $\mathcal{O}(n^5)$.

Hence the total time needed to compute all of these congruences is $\mathcal{O}(n^5)$. □

8. Decidability. Having shown that $\#\mathsf{CSP}$ has a dichotomy, we must consider whether it is effective. That is, given a relational structure $\mathfrak{G} = (D, \Gamma)$ can we decide algorithmically whether the problem $\#\mathsf{CSP}(\Gamma)$ is in FP or is $\#\mathsf{P}$-complete? This is the major question left open in [3]. Here we show that the answer is in the affirmative.

We will construct an algorithm to solve the following decision problem.

Strong Balance
Instance : A relational structure $\mathfrak{G} = (D, \Gamma)$.
Question : Is Γ strongly balanced?

Recall from Section 2 that we may assume that $\|\Gamma\| \geq q$. Thus, we may take $\|\Gamma\|$ as the measure of input size for Strong Balance. We bound the complexity of Strong Balance as a function of $\|\Gamma\|$. Complexity is a secondary issue, since $\|\Gamma\|$ is a constant in the nonuniform model for $\#\mathsf{CSP}(\Gamma)$. In the nonuniform model, we are only required to show that some algorithm exists to solve Strong Balance. However, we believe that the computational complexity of deciding the dichotomy is intrinsically interesting.

Our approach will be to show that the strong balance condition is equivalent to a structural property of Γ that can be checked in NP.

We must first verify that Γ is strongly rectangular, since otherwise it cannot be strongly balanced, by Lemma 29. Thus, we consider the following computational problem.

Strong Rectangularity
Instance : A relational structure $\mathfrak{S} = (D, \Gamma)$.
Question : Is Γ strongly rectangular?

Lemma 37. Strong Rectangularity *is in* NP.

Proof. We use the method of Lemma 8. We can verify that a given function φ is a Mal'tsev polymorphism in $\mathcal{O}(\|\Gamma\|^4)$ time. Thus, we select a function $\varphi\colon D^3 \to D$ nondeterministically in $\mathcal{O}(q^3) = \mathcal{O}(\|\Gamma\|^3)$ time and check that it is a Mal'tsev polymorphism in a further $\mathcal{O}(\|\Gamma\|^4)$ time. □

The remainder of this section is organised as follows. We first give definitions and notation that were held over from Section 2 because they are only used here. In Section 8.2, we give a characterisation of rank-one block matrices that we use in our decidability proof. The proof itself appears in Section 8.3.

8.1. Definitions and notation. An equivalent but different view of $\mathsf{CSP}(\Gamma)$ from the one we have used is often taken in the literature. This is to regard Φ as a finite structure with domain V and relations determined by the scopes of the constraints. Thus, we have relations $\tilde{H}$, where $(i_1, i_2, \ldots, i_r) \in \tilde{H}$ if $H(x_{i_1}, x_{i_2}, \ldots, x_{i_r})$ is a constraint. Then a satisfying assignment $\mathbf{x}$ is a *homomorphism* from Φ to Γ.

The following definitions and notation will be used in the remainder of this section. Let $[D_1 \to D_2]$ denote the set of functions from D_1 to D_2. Then a homomorphism between two relational structures $\mathfrak{S}_1 = (D_1, \Gamma_1)$, $\mathfrak{S}_2 = (D_2, \Gamma_2)$ is a function $\sigma \in [D_1 \to D_2]$ that preserves relations. Thus, for each r-ary relation $H_1 \in \Gamma_1$ there is a corresponding r-ary relation $H_2 \in \Gamma_2$ and, for each tuple $\mathbf{u} = (u_1, \ldots, u_r) \in H_1$, we have $\sigma(\mathbf{u}) = (\sigma(u_1), \ldots, \sigma(u_r)) \in H_2$. We will write $\sigma\colon \mathfrak{S}_1 \to \mathfrak{S}_2$ to indicate that σ is a homomorphism.

Let $[V \hookrightarrow D]$ denote the set of all *injective* functions $V \to D$ and let $[V \leftrightarrow D]$ denote the set of all *bijective* functions $V \to D$. If $\sigma\colon \mathfrak{S}_1 \to \mathfrak{S}_2$ and $\sigma \in [D_1 \hookrightarrow D_2]$, then σ is called a *monomorphism* and we will write $\sigma\colon \mathfrak{S}_1 \hookrightarrow \mathfrak{S}_2$. If σ is a bijective homomorphism and σ^{-1} is also a homomorphism, then σ is called an *isomorphism* and we write $\sigma\colon \mathfrak{S}_1 \leftrightarrow \mathfrak{S}_2$. Then $\mathfrak{S}_1$, $\mathfrak{S}_2$ are *isomorphic*, so isomorphic structures are the same up to relabelling. An *endomorphism* of a relational structure $\mathfrak{S}$ is a homomorphism $\sigma\colon \mathfrak{S} \to \mathfrak{S}$ and an *automorphism* is an isomorphism $\sigma\colon \mathfrak{S} \leftrightarrow \mathfrak{S}$. Note that the definition of an endomorphism is identical to that of a unary polymorphism. Note also that $[D \hookrightarrow D] = [D \leftrightarrow D]$, since D is finite, so an injective endomorphism is always an automorphism. Clearly, the identity function is always an automorphism, for any relational structure $\mathfrak{S}$.

We use the following construction of *powers of* $\mathfrak{S}$ (see, for example, [25, p. 282]).

For any relational structure $\mathfrak{S} = (D, \Gamma)$ and $k \in \mathbb{N}$, the relational structure $\mathfrak{S}^k = (D^k, \Gamma^k)$ is defined as follows. The domain is the Cartesian power D^k. The constraint language Γ^k is such that, for each r-ary relation $H \in \Gamma$, there is an r-ary $H^k \in \Gamma^k$, which is defined to be the following relation. If $\mathbf{u}_i = (u_{i,1}, u_{i,2}, \dots, u_{i,k}) \in D^k$ $(i \in [r])$, then $(\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_r) \in H^k$ if, and only if, $(u_{1,j}, u_{2,j}, \dots, u_{r,j}) \in H$ for all $j \in [k]$. Now, if Ψ is a pp-formula in Γ, we define the corresponding formula Ψ^k to be identical to Ψ, except that each occurrence of $H \in \Gamma$ is replaced by the corresponding relation $H^k \in \Gamma^k$. Observe that the relation Ψ^k is actually pp-definable in Γ, by the formula $\Psi^k(\mathbf{x}) = \Psi(\mathbf{x}_1) \wedge \Psi(\mathbf{x}_2) \wedge \cdots \wedge \Psi(\mathbf{x}_k)$, where $\mathbf{x}_i$ $(i \in [k])$ are disjoint n-tuples of variables. In particular, we have $|\Psi^k| = |\Psi|^k$.

Using this construction, the definition of a polymorphism can be reformulated. In this view of $\mathsf{CSP}(\Gamma)$, it follows directly that a k-ary polymorphism is just a homomorphism $\psi \colon \mathfrak{S}^k \to \mathfrak{S}$.

8.2. Rank-one block matrices. In our decidability proof, we use a different characterisation of rank-one block matrices, given by Corollary 40. This may seem more complicated than the original definition but it is more suited to our purpose.

Lemma 38. *A matrix A is a rank-one block matrix if, and only if, every 2×2 submatrix of A is a rank-one block matrix.*

Proof. Let A be a $k \times \ell$ rank-one block matrix and let

$$B = \begin{bmatrix} a_{ir} & a_{is} \\ a_{jr} & a_{js} \end{bmatrix} \qquad (i, j \in [k],\ i \neq j;\ r, s \in [\ell],\ r \neq s).$$

be any 2×2 submatrix of A. If any of a_{ir}, a_{is}, a_{jr}, a_{js} is zero, at least two must be zero, since A is rectangular. Then B is clearly a rank-one block matrix. If a_{ir}, a_{is}, a_{jr}, a_{js} are all nonzero, B must be a submatrix of some block of A. Since this block has rank one, B also has rank one.

Conversely, suppose A is not a rank-one block matrix. If its underlying relation is not rectangular, there exist $a_{ir}, a_{is}, a_{jr} > 0$ with $a_{js} = 0$. The corresponding matrix B clearly has rank 2, but has only one block so is not a rank-one block matrix. If the underlying relation of A is rectangular, then A must have a block of rank at least 2. This block must have some 2×2 submatrix B with rank 2 and all its elements $a_{ir}, a_{is}, a_{jr}, a_{js} > 0$. □

Lemma 39. *A rectangular 2×2 matrix A is a rank-one block matrix if, and only if, $a_{11}^2 a_{22}^2 a_{12} a_{21} = a_{12}^2 a_{21}^2 a_{11} a_{22}$.*

Proof. This equation holds if any of a_{11}, a_{22}, a_{12} or a_{21} is zero. But then rectangularity implies that at least two of them must be zero and A is a rank-one block matrix in all possible cases. Otherwise, the equation is equivalent to $a_{11}a_{22} = a_{12}a_{21}$, which is the condition that A is singular. So A is one block, with rank one. The argument is clearly reversible. □

Corollary 40. *A rectangular $k \times \ell$ matrix A is a rank-one block matrix if, and only if, $a_{ir}^2 a_{js}^2 a_{is} a_{jr} = a_{is}^2 a_{jr}^2 a_{ir} a_{js}$ for all $i, j \in [k]$ and all $r, s \in [\ell]$.*

Proof. When $i = j$ or $r = s$, the two sides of this equation are identical. Otherwise, the equality follows directly from Lemmas 38 and 39. □

Remark 9. It is possible to modify the above so that Corollary 40 involves products of only five elements, rather than six, but we do not pursue this refinement.

8.3. Decidability. To show the decidability of strong balance, we relax the criterion of strong balance, by noting the conditions sufficient for the success of the algorithm in Section 7.1. Observe that only ternary relations on $D \times D \times D^i$, for

$i \in [n-2]$, are required to be balanced. Therefore, let $\Psi(\mathbf{x})$, with $\mathbf{x} = (x_1, \ldots, x_n)$, be an arbitrary formula pp-definable in Γ, which we consider fixed for the rest of this section. Then, for the algorithm to succeed, it suffices that the $q \times q$ matrix

$$M(a,b) \;=\; \bigl|\{\mathbf{x} \in [V \to D] : \mathbf{x} \in \Psi,\ x_1 = a,\ x_2 = b\}\bigr| \qquad (\forall a, b \in D)$$

is always a rank-one block matrix. Note that we can always assume that the underlying relation of M is rectangular, since Γ is known to be strongly rectangular.

Remark 10. Call this condition *almost-strong* balance. It is equivalent to strong balance if $\mathsf{FP} \neq \#\mathsf{P}$. If $\mathfrak{S}$ is strongly balanced, it is clearly almost-strongly balanced. Almost-strong balance implies that the algorithm of Section 7.1 succeeds, which implies that $\#\mathsf{CSP}(\Gamma) \in \mathsf{FP}$. Thus $\#\mathsf{CSP}(\Gamma)$ is not $\#\mathsf{P}$-complete, which implies that it is strongly balanced by Lemma 31. This chain of implications requires $\mathsf{FP} \neq \#\mathsf{P}$, so we make that assumption in the remainder of this section. If $\mathsf{FP} = \#\mathsf{P}$, no dichotomy exists and the property of strong balance ceases to be of computational interest.

We may therefore take almost-strong balance as the criterion for strong balance. By Corollary 40, the condition for M to be a rank-one block matrix is that

$$M(a,c)^2 M(a,d) M(b,d)^2 M(b,c) \;=\; M(a,d)^2 M(a,c) M(b,c)^2 M(b,d), \tag{5}$$

for all $a, b, c, d \in D$.

We can reformulate the condition for strong balance using the construction of powers of $\mathfrak{S}$. If $\mathbf{a} = (a_1, \ldots, a_k)$ and $\mathbf{b} = (b_1, \ldots, b_k)$, the balance matrix M_k for Ψ^k is the $q^k \times q^k$ matrix

$$\begin{aligned} M_k(\mathbf{a}, \mathbf{b}) \;&=\; \bigl|\{\mathbf{x} \in [V \to D^k] \,:\, \mathbf{x} \in \Psi^k,\ x_1 = \mathbf{a}, x_2 = \mathbf{b}\}\bigr| \\ &=\; M(a_1, b_1) M(a_2, b_2) \cdots M(a_k, b_k)\,. \end{aligned}$$

Using this, equation (5) can be rewritten as

$$M_6(\bar{a}, \bar{c}) \;=\; M_6(\bar{a}, \bar{d})\,, \tag{6}$$

where

$$\bar{a} = (a,a,a,b,b,b),\ \ \bar{c} = (c,c,d,d,d,c),\ \ \bar{d} = (d,d,c,c,c,d)\,. \tag{7}$$

Fix $\bar{a}$, $\bar{c}$, $\bar{d}$ and, for notational simplicity, write $\bar{\mathfrak{S}}$ for $\mathfrak{S}^6$, $\bar{\Gamma}$ for Γ^6, $\bar{\Psi}$ for Ψ^6, $\bar{M}$ for M_6 and $\bar{D}$ for D^6. Then, from (6), we must verify that $\bar{M}(\bar{a}, \bar{c}) = \bar{M}(\bar{a}, \bar{d})$ for all relations $\bar{\Psi}$ which are pp-definable in $\bar{\Gamma}$ and given $\bar{a}, \bar{c}, \bar{d} \in \bar{D}$. We use a method of Lovász [24]; see also [15]. For $\bar{s} \in \bar{D}$, let

$$\begin{aligned} \mathrm{Hom}_{\bar{a},\bar{s}}(\bar{\Psi}) &= \{\mathbf{x} \in [V \to \bar{D}] : \mathbf{x} \in \bar{\Psi},\ x_1 = \bar{a},\ x_2 = \bar{s}\} \\ \mathrm{hom}_{\bar{a},\bar{s}}(\bar{\Psi}) &= |\mathrm{Hom}_{\bar{a},\bar{s}}(\bar{\Psi})|\,. \end{aligned}$$

However, a homomorphism $V \to \bar{D}$ that is consistent with $\bar{\Psi}$ is just a satisfying assignment to $\bar{\Psi}$. $\bar{M}(\bar{a}, \bar{s})$ is the number of such assignments with $x_1 = \bar{a}$ and $x_2 = \bar{s}$, i.e., the number of homomorphisms that map $x_1 \mapsto \bar{a}$ and $x_2 \mapsto \bar{s}$. This proves the following.

Lemma 41. *Γ is strongly balanced if, and only if,* $\mathrm{hom}_{\bar{a},\bar{c}}(\bar{\Psi}) = \mathrm{hom}_{\bar{a},\bar{d}}(\bar{\Psi})$ *for all formulae $\bar{\Psi}$ and all $\bar{a}, \bar{c}, \bar{d}$ of the form above.*

We will also need to consider the injective functions in $\mathrm{Hom}_{\bar{a},\bar{s}}(\bar{\Psi})$. For $\bar{s} \in \bar{D}$, let

$$\mathrm{Mon}\_\bar{a}, eus(\bar{\Psi}) = \{\mathbf{x} \in [V \hookrightarrow \bar{D}] : \mathbf{x} \in \bar{\Psi},\ x_1 = \bar{a},\ x_2 = \bar{s}\}$$

$$\mathrm{mon}_{\bar a,\bar s}(\bar\Psi) = |\mathrm{Mon}_{\bar a,\bar s}(\bar\Psi)|\,.$$

LEMMA 42. *$\mathrm{hom}_{\bar a,\bar c}(\bar\Psi) = \mathrm{hom}_{\bar a,\bar d}(\bar\Psi)$ for all $\bar\Psi$ if, and only if, $\mathrm{mon}_{\bar a,\bar c}(\bar\Psi) = \mathrm{mon}_{\bar a,\bar d}(\bar\Psi)$ for all $\bar\Psi$.*

Proof. Consider the set $\mathcal{I}$ of all partitions I of V into disjoint classes $\bar I_1,\dots,\bar I_{k_I}$, such that $1\in\bar I_1$, $2\in\bar I_2$. Writing $I\preceq I'$ whenever I is a refinement of I', $\mathbb{P}=(\mathcal{I},\preceq)$ is a poset. We will write $\perp$ for the partition into singletons, so $\perp\preceq I$ for all $I\in\mathcal{I}$.

Let V/I denote the set of classes $\bar I_1,\dots,\bar I_{k_I}$ of the partition I, so $|V/I|=k_I$, and let $\bar I_1$, $\bar I_2$ be denoted by $1/I$, $2/I$. Let $\bar\Psi/I$ denote the relation obtained from $\bar\Psi$ by imposing equality on all pairs of variables that occur in the same partition of I. Thus, the constraints $x_1=\bar a$, $x_2=\bar s$ become $x_{1/I}=\bar a$, $x_{2/I}=\bar s$. Then we have

$$\mathrm{hom}_{\bar a,\bar s}(\bar\Psi) \;=\; \mathrm{hom}_{\bar a,\bar s}(\bar\Psi/\perp) \;=\sum_{I\in\mathcal{I}}\mathrm{mon}_{\bar a,\bar s}(\bar\Psi/I) \;=\sum_{I\in\mathcal{I}}\mathrm{mon}_{\bar a,\bar s}(\bar\Psi/I)\zeta(\perp,I)\,, \tag{8}$$

where $\zeta(I,I')=1$, if $I\preceq I'$, and $\zeta(I,I')=0$, otherwise, is the ζ-function of the poset $\mathbb{P}$. Thus, if $\mathrm{mon}_{\bar a,\bar c}(\bar\Psi)=\mathrm{mon}_{\bar a,\bar d}(\bar\Psi)$ for all $\bar\Psi$, then

$$\mathrm{hom}_{\bar a,\bar c}(\bar\Psi) \;=\sum_{I\in\mathcal{I}}\mathrm{mon}_{\bar a,\bar c}(\bar\Psi/I)\zeta(\perp,I) \;=\sum_{I\in\mathcal{I}}\mathrm{mon}_{\bar a,\bar d}(\bar\Psi/I)\zeta(\perp,I) \;=\; \mathrm{hom}_{\bar a,\bar d}(\bar\Psi)\,. \tag{9}$$

More generally, the reasoning used to give (8) implies that

$$\mathrm{hom}_{\bar a,\bar s}(\bar\Psi/I) \;=\sum_{I\preceq I'}\mathrm{mon}_{\bar a,\bar s}(\bar\Psi/I') \;=\sum_{I'\in\mathcal{I}}\mathrm{mon}_{\bar a,\bar s}(\bar\Psi/I')\zeta(I,I')\,.$$

Now, Möbius inversion for posets [30, Ch. 25] implies that the matrix $\zeta\colon\mathcal{I}\times\mathcal{I}\to\{0,1\}$ has an inverse $\mu\colon\mathcal{I}\times\mathcal{I}\to\mathbb{Z}$. It follows directly that

$$\mathrm{mon}_{\bar a,\bar s}(\bar\Psi) \;=\; \sum_{I\in\mathcal{I}}\mathrm{hom}_{\bar a,\bar s}(\bar\Psi/I)\mu(\perp,I)\,.$$

Thus, if $\mathrm{hom}_{\bar a,\bar c}(\bar\Psi)=\mathrm{hom}_{\bar a,\bar d}(\bar\Psi)$ for all $\bar\Psi$, then

$$\mathrm{mon}_{\bar a,\bar c}(\bar\Psi) \;=\sum_{I\in\mathcal{I}}\mathrm{hom}_{\bar a,\bar c}(\bar\Psi/I)\mu(\perp,I) \;=\sum_{I\in\mathcal{I}}\mathrm{hom}_{\bar a,\bar d}(\bar\Psi/I)\mu(\perp,I) \;=\; \mathrm{mon}_{\bar a,\bar d}(\bar\Psi)\,. \tag{10}$$

Now, (9) and (10) give the conclusion. □

LEMMA 43. *$\mathrm{mon}_{\bar a,\bar c}(\bar\Psi)=\mathrm{mon}_{\bar a,\bar d}(\bar\Psi)$, for all $\bar\Psi$, if, and only if, there is an automorphism $\eta\colon\bar D\leftrightarrow\bar D$ of $\bar{\mathfrak{S}}=(\bar D,\bar\Gamma)$ such that $\eta(\bar a)=\bar a$ and $\eta(\bar c)=\bar d$.*

Proof. The condition holds if $\bar{\mathfrak{S}}$ has such an automorphism since, if $\bar\Psi(\mathbf{x}) = \exists\mathbf{y}\,\bar\Phi(\mathbf{x},\mathbf{y})$ for some $\bar\Phi$, then

$$\begin{aligned}
\mathrm{mon}_{\bar a,\bar c}(\bar\Psi) &= |\{\mathbf{x}\in[V\hookrightarrow\bar D] : x_1=\bar a,\ x_2=\bar c,\ \exists\mathbf{y}\,(\mathbf{x},\mathbf{y})\in\bar\Phi\}|\\
&= |\{\eta(\mathbf{x})\in[V\hookrightarrow\bar D] : x_1=\eta(\bar a),\ x_2=\eta(\bar c),\ \exists\mathbf{y}\,(\eta(\mathbf{x}),\eta(\mathbf{y}))\in\bar\Phi\}|\\
&= |\{\mathbf{x}\in[V\hookrightarrow\bar D] : x_1=\bar a,\ x_2=\bar d,\ \exists\mathbf{y}\,(\mathbf{x},\mathbf{y})\in\bar\Phi\}|\\
&= \mathrm{mon}_{\bar a,\bar d}(\bar\Psi)\,.
\end{aligned}$$

For the converse, suppose we have $\mathrm{mon}_{\bar a,\bar c}(\bar\Psi)=\mathrm{mon}_{\bar a,\bar d}(\bar\Psi)$ for all $\bar\Psi$. Consider the following $\bar\Gamma$-formula $\bar\Phi$ with domain $\bar D$ and variables x_i $(i\in\bar D)$,

$$\bar\Phi(\mathbf{x}) \;=\; \bigwedge_{\bar H\,\in\,\bar\Gamma}\;\bigwedge_{(\bar u_1,\dots,\bar u_r)\,\in\,\bar H}\bar H(x_{\bar u_1},\dots,x_{\bar u_r})\,.$$

Then

$$\mathrm{Mon}_{\bar a,\bar s}(\bar\Phi) \;=\; \{\mathbf{x} \in [\bar D \hookrightarrow \bar D] :\; x_{\bar a} = \bar a,\, x_{\bar c} = \bar s,\, \mathbf{x} \in \bar\Phi\}\,.$$

We have $\mathrm{Mon}_{\bar a,\bar c}(\bar\Phi) \neq \emptyset$, since the identity assignment $x_i = i$ $(i \in \bar D)$ is clearly satisfying. Thus, by the assumption, $\mathrm{Mon}_{\bar a,\bar d}(\bar\Phi) \neq \emptyset$. Let $\eta \in \mathrm{Mon}_{\bar a,\bar d}(\bar\Phi)$, so η is an endomorphism of $\bar{\mathfrak{G}}$ with $\eta(\bar a) = \bar a$, $\eta(\bar c) = \bar d$. Since $[D \hookrightarrow D] = [D \leftrightarrow D]$, $\eta\colon D \leftrightarrow D$ is the required automorphism. □

Corollary 44. *$\mathfrak{G} = (D, \Gamma)$ is strongly balanced if, and only if, for all $a, b, c, d \in D$ and $\bar a, \bar c, \bar d$ as defined in* (7)*, $\bar{\mathfrak{G}} = (\bar D, \bar\Gamma)$ has an automorphism η such that $\eta(\bar a) = \bar a$ and $\eta(\bar c) = \bar d$.*

Proof. This follows from (6) and Lemmas 41, 42 and 43. □

This characterisation of strong balance leads to a nondeterministic algorithm.

Theorem 45. Strong Balance *is in* NP.

Proof. We first determine whether Γ is strongly rectangular, using the method of Lemma 37. If it is not, then Γ is not strongly rectangular by Lemma 29.

Otherwise, we can construct $\bar{\mathfrak{G}} = (\bar D, \bar\Gamma)$ in time $\mathcal{O}(\|\Gamma\|^6)$. Let $\bar q = q^6 = |\bar D|$ and let Π denote the set of $\bar q!$ permutations of $\bar D$. Each $\pi \in \Pi$ is a function $\pi\colon \bar D \hookrightarrow \bar D$ and so a potential automorphism of $\bar{\mathfrak{G}}$. For each of the q^4 possible choices $a, b, c, d \in D$, we determine $\bar a, \bar c, \bar d \in \bar D$ in polynomial time. We select $\pi \in \Pi$ nondeterministically and check that $\pi(\bar a) = \bar a$, $\pi(\bar c) = \bar d$ and that π preserves all $\bar H \in \bar\Gamma$. The computation requires $\mathcal{O}(q^4\|\bar\Gamma\|^2) = \mathcal{O}(\|\Gamma\|^{16})$ time in total, so everything other than the $\mathcal{O}(q^{10}) = \mathcal{O}(\|\Gamma\|^{10})$ nondeterministic choices can be done deterministically in a polynomial number of steps. □

Remark 11. We have paid little attention to the efficiency of the computations in Theorem 45. If the elements of D are encoded as binary numbers in $[q]$, comparisons and nondeterministic choices require $\mathcal{O}(\log q)$ bit operations, rather than the $\mathcal{O}(1)$ operations in our accounting. On the other hand, membership in H^6 can be tested in $\mathcal{O}(\|H\|)$ comparisons, rather than the $\mathcal{O}(\|H\|^6)$ that we have allowed. This might be reduced further by storing H in a suitable data structure, instead of a simple matrix. We could also use Remark 9 to improve the algorithm of Theorem 45.

Remark 12. Theorem 45 and Lemma 32 together imply that the following problem, posed by Bulatov [3], can also be decided in NP.

Congruence Singularity
Instance : A relational structure $\mathfrak{G} = (D, \Gamma)$.
Question : Is Γ congruence singular?

Whether this can be shown directly, and not via Strong Balance, remains an open question.

9. Conclusions.

We have shown that there is an effective dichotomy for the whole of #CSP. We have given a new, and simpler, proof for the existence of the dichotomy and the first proof of its decidability.

The complexity of our counting algorithm is $\mathcal{O}(n^5)$, whereas algorithms for most known counting dichotomies are of lower complexity, often $\mathcal{O}(n)$. Can the complexity of the general algorithm be improved to $\mathcal{O}(n^4)$, or better? Since frames, on which the algorithm is based, have size $\mathcal{O}(n)$, there is no obvious reason why this cannot be done.

A second problem that we have not yet considered is an extension to a dichotomy for *weighted* counting problems [8, 14]. We believe that this is possible. In fact, a dichotomy for *rational* weights has already been shown in [7]. This gives an indirect argument, using the unweighted dichotomy. Decidability of the dichotomy of [7] now follows from Section 8 of this paper.

A third issue is to investigate whether known counting dichotomies can be recovered from these general theorems. We have some preliminary results in this direction. The characterisation of Lemma 43 appears to be useful in this respect.

A fourth problem is to determine the complexity of Strong Balance more precisely, rather than just establishing membership in NP. Strong Balance seems unlikely to be NP-complete as the automorphism tests required can be coded into a single instance of the graph isomorphism problem. However, it is not obvious whether the converse reduction is possible so it may be that Strong Balance is in P.

Finally, a deeper question that arises from our work is to what extent the detailed properties of the algebras associated with CSP instances are of real significance. In recent years, the algebraic approach has proven successful in the study of CSP, but it is possible that these algebras are more complicated objects than the relations they are intended to capture.

Note. Since this paper was written, Cai, Chen and Lu have extended and strengthened our methods to give an effective dichotomy for the weighted counting problem [9].

Acknowledgments. The authors are grateful to Jin-Yi Cai, Xi Chen and Andrei Krokhin for carefully reading drafts of an earlier version of this paper. We are also grateful to Andrei Bulatov for explaining parts of his proof, and to Leslie Ann Goldberg for useful discussions. We also thank a referee for pointing out the issue discussed in Remark 10.

REFERENCES

[1] A. A. Bulatov. A dichotomy theorem for constraint satisfaction problems on a 3-element domain. *Journal of the ACM*, 53(1):66–120, 2006.

[2] A. A. Bulatov. The complexity of the counting constraint satisfaction problem. *Electronic Colloquium on Computational Complexity*, 14(093), 2007. (Revised Feb. 2009).

[3] A. A. Bulatov. The complexity of the counting constraint satisfaction problem. In *Proc. 35th International Colloquium on Automata, Languages and Programming (Part 1)*, LNCS 5125, pp. 646–661. Springer, 2008.

[4] A. A. Bulatov and V. Dalmau. A simple algorithm for Mal'tsev constraints. *SIAM Journal on Computing*, 36(1):16–27, 2006.

[5] A. A. Bulatov and V. Dalmau. Towards a dichotomy theorem for the counting constraint satisfaction problem. In *Proc. 44th Annual IEEE Symposium on Foundations of Computer Science*, pp. 562–573, IEEE, 2003.

[6] A. A. Bulatov and V. Dalmau. Towards a dichotomy theorem for the counting constraint satisfaction problem. *Information and Computation*, 205(5):651–678, 2007.

[7] A. A. Bulatov, M. E. Dyer, L. A. Goldberg, M. Jalsenius, M. R Jerrum and D. Richerby. The complexity of weighted and unweighted #CSP. `arXiv`:1005.2678 [cs.CC], May 2010.

[8] A. A. Bulatov and M. Grohe. The complexity of partition functions. *Theoretical Computer Science*, 348(2–3):148–186, 2005.

[9] J.-Y. Cai, X. Chen and P. Lu, Non-negative weighted #CSPs: An effective complexity dichotomy, `arXiv`: 1012.5659 [cs.CC], December 2010.

[10] J.-Y. Cai, P. Lu, and M. Xia. Holant problems and counting CSP. In *Proc. 41st Annual ACM Symposium on Theory of Computing*, pp. 715–724. ACM, 2009.

[11] N. Creignou and M. Hermann. Complexity of generalized satisfiability counting problems. *Information and Computation*, 125(1):1–12, 1996.

[12] K. Denecke and S. L. Wismath. *Universal Algebra and Applications in Theoretical Computer Science.* Chapman and Hall/CRC, 2002.

[13] M. E. Dyer, L. A. Goldberg, and M. R. Jerrum. A complexity dichotomy for hypergraph partition functions. *Computational Complexity*, 19(4):605–633, 2010.

[14] M. E. Dyer, L. A. Goldberg, and M. R. Jerrum. The complexity of weighted Boolean #CSP. *SIAM Journal on Computing*, 38(5):1970–1986, 2009.

[15] M. E. Dyer, L. A. Goldberg, and M. S. Paterson. On counting homomorphisms to directed acyclic graphs. *Journal of the ACM*, 54(6), 2007.

[16] M. E. Dyer and C. S. Greenhill. The complexity of counting graph homomorphisms. *Random Structures and Algorithms*, 17(3–4):260–289, 2000. (Corrigendum in *Random Structures and Algorithms*, 25(3):346–352, 2004.).

[17] T. Feder and M. Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory. *SIAM Journal on Computing*, 28(1):57–104, 1998.

[18] R. Freese and R. McKenzie. *Commutator Theory for Congruence Modular Varieties.* Cambridge University Press, 1987.

[19] D. Geiger. Closed systems of functions and predicates. *Pacific Journal of Mathematics*, 27:95–100, 1968.

[20] P. Hell and J. Nešetřil. On the complexity of H-coloring. *Journal of Combinatorial Theory (Series B)*, 48(1):92–110, 1990.

[21] D. Hobby and R. McKenzie. *The Structure of Finite Algebras*, vol. 76 of *Contemporary Mathematics.* American Mathematical Society, 1988.

[22] P. G. Kolaitis and M. Y. Vardi. Conjunctive-query containment and constraint satisfaction. In *Proc. 17th ACM Symposium on Principles of Database Systems (PODS '98)*, pp. 205–213, New York, 1998. ACM.

[23] R. E. Ladner. On the structure of polynomial time reducibility. *Journal of the ACM*, 22(1):155–171, 1975.

[24] L. Lovász. Operations with structures. *Acta. Math. Acad. Sci. Hung.*, 18:321–328, 1967.

[25] J. Nešetřil, M. H. Siggers and L. Zádori. A combinatorial constraint satisfaction problem dichotomy classification conjecture. *European Journal of Combinatorics*, 31(1):280–296, 2010.

[26] T. Schaefer. The complexity of satisfiability problems. In *Proc. 10th Annual ACM Symposium on Theory of Computing*, pp. 216–226. ACM Press, 1978.

[27] S. Toda. On the computational power of PP and ⊕P. In *Proc. 30th Annual Symposium on Foundations of Computer Science*, pp. 514–519. IEEE Computer Society, 1989.

[28] L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201, 1979.

[29] L. G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.

[30] J. van Lint and R. Wilson. *A Course in Combinatorics* (2nd ed.). CUP, 2001.