

# Non-dichotomies in Constraint Satisfaction Complexity

Manuel Bodirsky<sup>1</sup> and Martin Grohe<sup>2</sup>

<sup>1</sup> École Polytechnique (CNRS), France

<sup>2</sup> Humboldt-Universität zu Berlin, Germany

**Abstract.** We show that every computational decision problem is polynomial-time equivalent to a constraint satisfaction problem (CSP) with an infinite template. We also construct for every decision problem  $L$  an  $\omega$ -categorical template  $\Gamma$  such that  $L$  reduces to  $\text{CSP}(\Gamma)$  and  $\text{CSP}(\Gamma)$  is in  $\text{coNP}^L$  (i.e., the class  $\text{coNP}$  with an oracle for  $L$ ). CSPs with  $\omega$ -categorical templates are of special interest, because the universal-algebraic approach can be applied to study their computational complexity.

Furthermore, we prove that there are  $\omega$ -categorical templates with  $\text{coNP}$ -complete CSPs and  $\omega$ -categorical templates with  $\text{coNP}$ -intermediate CSPs, i.e., problems in  $\text{coNP}$  that are neither  $\text{coNP}$ -complete nor in  $\text{P}$  (unless  $\text{P}=\text{coNP}$ ). To construct the  $\text{coNP}$ -intermediate CSP with  $\omega$ -categorical template we modify the proof of Ladner's theorem. A similar modification allows us to also prove a non-dichotomy result for a class of left-hand side restricted CSPs, which was left open in [10]. We finally show that if the so-called *local-global conjecture* for *infinite constraint languages* (over a finite domain) is false, then there is no dichotomy for the constraint satisfaction problem for infinite constraint languages.

## 1 Introduction

Let  $\Gamma$  be a relational structure over a finite signature  $\tau$ , also called *template* or *constraint language* in the following. The *constraint satisfaction problem (CSP)* of  $\Gamma$ , denoted by  $\text{CSP}(\Gamma)$ , is the computational problem to decide whether there exists a homomorphism from a given finite  $\tau$ -structure to  $\Gamma$ . For finite relational structures  $\Gamma$  it has been conjectured that the computational complexity of  $\text{CSP}(\Gamma)$  exhibits a *dichotomy*: it is in  $\text{P}$  or  $\text{NP}$ -complete [9, 6].

Ladner's theorem [14] states that there are computational problems in  $\text{NP}$  that are neither in  $\text{P}$  nor  $\text{NP}$ -complete (unless  $\text{P}=\text{NP}$ ); we also say that these problems are *NP-intermediate*. However, the problems constructed by Ladner are highly artificial, and the lack of natural computational problems that could be  $\text{NP}$ -intermediate is one of the phenomena in complexity theory that is not well understood. Even for well-studied problems in  $\text{NP}$  that are neither known to be in  $\text{P}$  nor known to be  $\text{NP}$ -hard, such as the Graph Isomorphism problem, there is usually no strong evidence that they are not in  $\text{P}$ . From the computational

complexity perspective, it would therefore be very interesting if there were NP-intermediate CSPs with finite domains (since CSPs for finite templates are, arguably, relatively natural computational problems).

The CSP dichotomy conjecture is wide open. Bulatov, Krokhin, and Jeavons [6] give a sufficient condition for a finite domain CSP to be NP-hard, and they conjecture that all other problems are in P. This conjecture is based on the so-called *universal-algebraic approach* to the CSP, and the mentioned condition is formulated in terms of an algebra that can be associated to the template  $\Gamma$ . Most work in the area goes into finding larger and larger tractable classes of CSPs [13, 3, 4]. But not all researchers in the area believe in the conjecture. On the negative side, Feder and Vardi [9] introduced several heavily restricted subclasses of NP—for example monadic strict NP (MSNP)—that do *not* have a dichotomy. In fact, every problem in NP has a polynomial-time equivalent problem in MSNP.

In this paper we are mainly interested in infinite templates  $\Gamma$ . A particularly interesting class of infinite templates is the class of  $\omega$ -categorical<sup>1</sup> structures, because  $\omega$ -categorical templates  $\Gamma$  generalize finite templates in such a way that the above mentioned universal-algebraic approach still applies. For example, it holds that the complexity of the CSP( $\Gamma$ ) is fully captured by the polymorphism clone of  $\Gamma$  (even by the pseudo-variety generated by the algebra of the polymorphism clone [2]).

## Results

We show that for every computational decision problem  $L$  there exists a polynomial-time equivalent constraint satisfaction problem with an infinite template  $\Gamma$ . This improves previous complexity results about infinite domain constraint satisfaction problems obtained by Bauslaugh [1] and Schwandtner [17]. We also construct an  $\omega$ -categorical template  $\Gamma$  such that  $L$  reduces to CSP( $\Gamma$ ) and CSP( $\Gamma$ ) is in  $\text{coNP}^L$  (i.e., the class  $\text{coNP}$  with an oracle for  $L$ ).

Furthermore, we prove that there are  $\omega$ -categorical templates with  $\text{coNP}$ -complete CSPs and  $\omega$ -categorical templates with *coNP-intermediate* CSPs, i.e., problems in  $\text{coNP}$  that are neither  $\text{coNP}$ -complete nor in P (unless  $\text{P}=\text{coNP}$ ). To show this we use templates that are *countable homogeneous directed graphs*. These graphs are  $\omega$ -categorical, and even though there are uncountably many non-isomorphic countable homogeneous directed graphs, they are model-theoretically well-understood [8]. In our construction we apply a modification of Ladner's proof technique. It remains open whether there are  $\omega$ -categorical structures  $\Gamma$  such that CSP( $\Gamma$ ) is NP-intermediate.

In another line of research, the complexity of the constraint satisfaction problem has been studied for restricted classes of input structures. Let  $\tau$  be a finite relational signature, and let  $\mathcal{C}$  and  $\mathcal{D}$  be two classes of finite  $\tau$ -structures. Then CSP( $\mathcal{C}, \mathcal{D}$ ) is the computational problem where the input consists of *two*

---

<sup>1</sup> A structure is  $\omega$ -categorical if it has for all  $k$  only a finite number of  $k$ -types (in the model-theoretic sense [12]).

structures  $C \in \mathcal{C}$  and  $D \in \mathcal{D}$ , and the question is whether there exists a homomorphism from  $C$  to  $D$ . It has been shown that when  $\mathcal{D}_0$  is the set of all finite  $\tau$ -structures, then for any recursively enumerable class of finite  $\tau$ -structures  $\mathcal{C}$  the problem  $\text{CSP}(\mathcal{C}, \mathcal{D}_0)$  is tractable if and only if  $\mathcal{C}$  has bounded tree-width modulo homomorphic equivalence [10]. Note that this is not a dichotomy theorem.

With a similar modification of Ladner's proof as in our previous result, we can show that there is an efficiently decidable class  $\mathcal{C}$  of finite (undirected) graphs such that  $\text{CSP}(\mathcal{C}, \mathcal{D}_0)$  is NP-intermediate (here,  $\mathcal{D}_0$  is the class of all finite undirected graphs); this was left open in [10]. In particular, this shows that there is no dichotomy for problems of the form  $\text{CSP}(\mathcal{C}, \mathcal{D})$ . The same result was independently obtained in a recent paper by Chen, Thurley, and Weyer [7].

Finally, we show a connection between the dichotomy conjecture for *infinite constraint languages over a finite domain* (implied by Conjecture 4.12 of [5]) and the so-called *local-global conjecture*. If  $\Gamma$  is a relational structure with an infinite signature and a finite domain, then  $\text{CSP}(\Gamma)$  is called *locally tractable* if  $\text{CSP}(\Gamma')$  is tractable for all reducts  $\Gamma'$  of  $\Gamma$  with a finite signature. Note that for the notion of local tractability we can fix an arbitrary representation of the relations in the input instances.  $\text{CSP}(\Gamma)$  is called *globally tractable* if there is a polynomial-time algorithm that solves  $\text{CSP}(\Gamma)$  where the relations in the input instances are represented by fully specifying the relation  $R \subset D(\Gamma)^k$ . Obviously, global tractability implies local tractability. The *local-global conjecture* says that  $\text{CSP}(\Gamma)$  is locally tractable if and only if it is globally tractable.

If the local-global conjecture is true, then the dichotomy for finite constraint languages implies the dichotomy for infinite constraint languages. We show that in a certain sense the converse is true as well: if the global tractability conjecture is false, then there is no dichotomy for infinite constraint languages. In other words, if the dichotomy conjecture for finite constraint languages is true, then local and global tractability are equivalent.

## Preliminaries

A *relational signature*  $\tau$  is a set of *relation symbols*  $R_i$ , each of which has an associated finite *arity*  $k_i$ . The signature will be finite unless stated otherwise. A *relational structure*  $\Gamma$  over the signature  $\tau$  (also called  $\tau$ -*structure*) consists of a set  $D_\Gamma$  (the *domain*) together with a relation  $R \subseteq D_\Gamma^{k_i}$  for each relation symbol of arity  $k_i$  from  $\tau$ . If the reference to the relational structure is clear from the context, we use for simplicity the same symbol for a relation symbol and the corresponding relation. If necessary, we write  $R^\Gamma$  for the relation  $R$  belonging to the structure  $\Gamma$ . We call the elements of  $D_\Gamma$  the *vertices* of  $\Gamma$ .

Let  $\Gamma$  and  $\Gamma'$  be  $\tau$ -structures. A *homomorphism* from  $\Gamma$  to  $\Gamma'$  is a function  $f$  from  $D_\Gamma$  to  $D_{\Gamma'}$  such that for each  $n$ -ary relation symbol  $R$  in  $\tau$  and each  $n$ -tuple  $(a_1, \dots, a_n)$ , if  $(a_1, \dots, a_n) \in R^\Gamma$ , then  $(f(a_1), \dots, f(a_n)) \in R^{\Gamma'}$ . In this case we say that the map  $f$  *preserves* the relation  $R$ . Injective homomorphisms that also preserve the complement of each relation are called *embeddings*. Surjective embeddings are called *isomorphisms*, and isomorphisms between  $\Gamma$  and  $\Gamma'$  are called *automorphisms*.

The *order* of a structure  $\Gamma$ , denoted by  $|\Gamma|$ , is the cardinality of  $D_\Gamma$ . An *induced substructure* of a structure  $\Gamma$  is a structure  $\Gamma'$  with  $D_{\Gamma'} \subseteq D_\Gamma$  and  $R^{\Gamma'} = R^\Gamma \cap D_{\Gamma'}^n$ , for each  $n$ -ary  $R \in \tau$ . The *union* of two  $\tau$ -structures  $\Gamma, \Gamma'$  is the structure  $\Gamma \cup \Gamma'$  with domain  $D_\Gamma \cup D_{\Gamma'}$  and relations  $R^{\Gamma \cup \Gamma'} = R^\Gamma \cup R^{\Gamma'}$  for all  $R \in \tau$ . The intersection  $\Gamma \cap \Gamma'$  is defined similarly. A *disjoint union* of  $\Gamma$  and  $\Gamma'$  is the union of isomorphic copies of  $\Gamma$  and  $\Gamma'$  with disjoint domains. As disjoint unions are unique up to isomorphism, we usually speak of *the* disjoint union of  $\Gamma$  and  $\Gamma'$ . A structure is called *connected* if it is not the disjoint union of two nonempty structures. For a mapping  $f$  defined on the domain of a  $\tau$ -structure  $\Gamma$ , we let  $f(\Gamma)$  be the  $\tau$ -structure with domain  $f(D_\Gamma)$  and relations  $R^{f(\Gamma)} = \{(f(a_1), \dots, f(a_n)) \mid (a_1, \dots, a_n) \in R^\Gamma\}$  for every  $n$ -ary  $R \in \tau$ .

Classes of structures are always assumed to be closed under isomorphism.

## 2 Templates of All Complexities

In this section we show that for every computational decision problem there exists a polynomial-time equivalent constraint satisfaction problem with an infinite template  $\Gamma$ . Previously, it was known that for every recursive function  $f$  there exists an infinite structure  $\Gamma$  such that  $\text{CSP}(\Gamma)$  is decidable, but has time complexity at least  $f$  (a result due to Bauslaugh [1]). Recently, Schwandtner gave upper and lower bounds in the exponential time hierarchy for some infinite domain CSPs [17]; but these bounds leave an exponential gap.

**Theorem 1.** *Let  $L$  be a language over a finite alphabet  $\Sigma$ . Then there is an infinite relational structure  $\Gamma$  such that  $L$  is polynomial-time Turing equivalent to  $\text{CSP}(\Gamma)$ .*

*Proof (Idea).* We construct an infinite relational structure  $\Gamma$  over the signature  $\tau$  that contains pairwise distinct unary relation symbols  $P_a$  for all elements of  $a \in \Sigma$ ; moreover,  $\tau$  contains a binary relation symbol  $N$  and unary relation symbols  $S$  and  $T$ .

For each word  $w \in \Sigma^*$  let  $W$  be the following  $\tau$ -structure with vertices  $1, \dots, |w|$ . The relation  $N^W$  is  $\{(i, i + 1) \mid 1 \leq i < |w|\}$ . The unary relation symbol  $P_a^W$  holds on  $j \in \{1, \dots, |w|\}$  iff the  $j$ th symbol in  $w$  is  $a$ . Finally,  $S^W = \{1\}$  and  $T^W = \{|w|\}$ . Let  $\mathcal{A}$  be  $\{W \mid w \in L\}$ .

Let  $\mathcal{X}$  be the set of  $\tau$ -structures with domain  $\{1, \dots, n\}$ , for some  $n$ , where the symbol  $N$  is interpreted by the relation  $\{(i, i + 1) \mid 1 \leq i < n\}$ , each vertex  $i$  is contained in at most one relation  $P_a$ , for  $a \in \Sigma$ , and at least one of the following conditions is satisfied:

- $S$  holds for none of the elements  $1, \dots, n$ .
- $T$  holds for none of the elements  $1, \dots, n$ .
- There is an element from  $1, \dots, n$  such that for all  $a \in \Sigma$  the relation  $P_a$  does not hold.

The structure  $\Gamma$  is the infinite disjoint union over all structures in  $\mathcal{A} \cup \mathcal{X}$ . The proof that  $\Gamma$  has the required properties has been omitted due to space restrictions. □

### 3 $\omega$ -Categorical Templates of Various Complexities

A relational structure is called  $\omega$ -categorical if for all  $k \geq 1$  there are at most finitely many inequivalent first-order formulas with  $k$  free variables. This definition of  $\omega$ -categoricity is by the theorem of Ryll-Nardzewski equivalent to the standard definition of  $\omega$ -categoricity [12]. With the definition given here it is easy to see that the structure  $\Gamma$  constructed in Section 2 is *not*  $\omega$ -categorical, since there are infinitely many inequivalent first-order formulas with two free variables  $x$  and  $y$  that express that  $x$  and  $y$  are at distance  $k$  with respect to the relation  $N$ .

In this section we study the computational complexity of  $\text{CSP}(\Gamma)$  if  $\Gamma$  is an  $\omega$ -categorical structure, and prove the following.

**Theorem 2.** *For every language  $L$  over a finite signature  $\Sigma$  there exists an  $\omega$ -categorical structure  $\Gamma$  such that  $L$  reduces to  $\text{CSP}(\Gamma)$  and  $\text{CSP}(\Gamma)$  is in  $\text{coNPL}$ .*

For constructing  $\omega$ -categorical templates, we need a few preliminaries from model theory. Let  $B_1, B_2$  be  $\tau$ -structures such that  $A = B_1 \cap B_2$  is an induced substructure of both  $B_1$  and  $B_2$ . Then we call  $B_1 \cup B_2$  the *free amalgam* of  $B_1, B_2$  over  $A$ . More generally, a  $\tau$ -structure  $C$  is an *amalgam of  $B_1$  and  $B_2$  over  $A$*  if for  $i = 1, 2$  there are embeddings  $f_i$  of  $B_i$  to  $C$  such that  $f_1(a) = f_2(a)$  for all  $a \in D_A$ . Recall that classes of structures are always assumed to be closed under isomorphism. A class  $\mathcal{A}$  of  $\tau$ -structures has the *amalgamation property* if for all  $A, B_1, B_2 \in \mathcal{A}$  with  $A = B_1 \cap B_2$  there is a  $C \in \mathcal{A}$  that is an amalgam of  $B_1$  and  $B_2$  over  $A$ . A class  $\mathcal{K}$  of finite  $\tau$ -structures that has the amalgamation property and is closed under taking induced substructures is called an *amalgamation class*.

The following basic result is known as Fraïssé's Theorem in model theory (see Theorem 6.1.2 in [12]):

**Fact 3.** *Let  $\mathcal{K}$  be an amalgamation class. Then there is an  $\omega$ -categorical  $\tau$ -structure  $\Gamma$  such that  $\mathcal{K}$  is the class of finite induced substructures of  $\Gamma$ .*

*The structure  $\Gamma$ , which is unique up to isomorphism, is called the Fraïssé limit of  $\mathcal{K}$ .*

A few remarks are necessary to relate our version of the theorem to the more general version stated, for example, in [12]: Firstly, the amalgamation property is usually defined in a slightly more complicated way where the structure  $A$  is not necessarily an induced substructure of the  $B_i$ , but embedded into  $B_i$ . As we assume classes of structures to be closed under isomorphism, this makes no difference. Secondly, the class  $\mathcal{A}$  in Fraïssé's Theorem usually has to have another property known as the *joint embedding property*. However, for relational structures the joint embedding property is subsumed by the amalgamation property. And thirdly, in general the Fraïssé limit  $\Gamma$  is *homogeneous*, but not necessarily  $\omega$ -categorical. But for finite relational vocabularies, homogeneity implies  $\omega$ -categoricity.

Let us now turn to the proof of Theorem 2. We encode words over the alphabet  $\Sigma$  by structures similarly, but not exactly as in the proof of Theorem 1. Let  $\tau$  be

a signature that contains the binary relation symbols  $N, \neq$ , the unary relation symbols  $S, T$ , and a unary relation symbol  $P_a$  for each  $a \in \Sigma$ . With each word  $w = a_1 \dots a_n \in \Sigma^*$  we associate the  $\tau$ -structure  $W$  with universe  $\{0, 1, \dots, n+1\}$ ,  $N^W = \{(i, i+1) \mid 0 \leq i < n+1\}$ ,  $\neq^W = \{(i, j) \mid 0 \leq i, j \leq n+1, i \neq j\}$ ,  $S^W = \{0\}$ ,  $T^W = \{n+1\}$ , and  $P_a = \{i \mid 1 \leq i \leq n, a_i = a\}$ .

For every  $\tau$ -structure  $A$ , we define an undirected graph  $G_N(A)$  to be the graph with vertex set  $D_A$  and an edge between  $a, b \in D_A$  if and only if  $a \neq b$  and  $(a, b) \in N^A$  or  $(b, a) \in N^A$ . We say that  $A$  is *connected by  $N$*  if the graph  $G_N(A)$  is connected.

Let  $\mathcal{C}$  be the class of all  $\tau$ -structures isomorphic to a structure  $W$  for some  $w \in \Sigma^* \setminus L$ . Let  $\mathcal{K}$  be the class of all  $\tau$ -structures  $A$  with the following properties.

- (1) The binary relation  $\neq^A$  is *anti-reflexive*, i.e., it does not contain pairs of the form  $(x, x)$ .
- (2) The relations  $S^A, T^A$ , and  $P_a$  for  $a \in \Sigma$  are pairwise disjoint.
- (3)  $N^A$  is anti-reflexive and anti-symmetric. Furthermore, if  $\neq^A$  has cardinality  $|A|(|A| - 1)$ , then the graph  $G_N(A)$  is acyclic.
- (4)  $A$  does not contain a structure from  $\mathcal{C}$  as an induced substructure.

**Lemma 1.** *The class  $\mathcal{K}$  is an amalgamation class.*

*Proof.* It is straightforward to verify that  $\mathcal{K}$  is closed under isomorphisms and induced substructures. To show that  $\mathcal{K}$  has the amalgamation property, let  $B_1, B_2 \in \mathcal{K}$  such that  $A = B_1 \cap B_2$  is an induced substructure of both  $B_1$  and  $B_2$ . We claim that the free amalgam  $C = B_1 \cup B_2$  is contained in  $\mathcal{K}$ .

We have to prove that  $C$  has properties (1)–(4) from page 189. This is obvious for properties (1) and (2), because both  $B_1$  and  $B_2$  have the property and their intersection  $A$  is an induced substructure of both structures. To see that  $C$  has property (3), note that either  $C = B_1$  or  $C = B_2$ , and  $C$  inherits the property from the respective structure, or  $D_C \setminus D_{B_1} \neq \emptyset$  and  $D_C \setminus D_{B_2} \neq \emptyset$ , and  $\neq^C = \neq^{B_1} \cup \neq^{B_2}$  has cardinality less than  $|C|(|C| - 1)$ . It remains to prove that  $C$  has property (4). Suppose for contradiction that  $C$  has an induced substructure  $W \in \mathcal{C}$ . As  $\neq^W$  connects all pairs of distinct vertices of  $W$ , the structure  $W$  must be an induced substructure of  $B_1$  or of  $B_2$ , which is a contradiction.

Let  $\Gamma$  be the Fraïssé-limit of  $\mathcal{K}$ .

**Lemma 2.** *A finite structure  $A$  has a homomorphism to  $\Gamma$  if and only if  $A$  has properties (1)–(4).*

*Proof (of Theorem 2).* There is the following reduction from  $L$  to  $\text{CSP}(\Gamma)$ . Given a word  $w \in \Sigma^*$ , let  $W$  be the corresponding  $\tau$ -structure. If  $w \notin L$ , then  $W$  is in  $\mathcal{C}$ , and hence  $W$  does not satisfy 4. Lemma 2 then implies that there is no homomorphism from  $W$  to  $\Gamma$ . If  $w \in L$ , then  $W \notin \mathcal{C}$ . The structure  $W$  has no induced substructure from  $\mathcal{C}$ , because every proper induced substructure of  $W$  either does not have an element in  $S$ , or does not have an element in  $T$ , or is not connected by  $N$ . Therefore,  $W$  homomorphically maps to  $\Gamma$ . This shows

that the function that returns for a given word  $w$  its word-structure  $W$  is a polynomial-time many-one reduction from  $L$  to  $\text{CSP}(\Gamma)$ .

We finally show that  $\text{CSP}(\Gamma)$  can be decided by a universal-nondeterministic polynomial-time algorithm with an oracle for  $L$ , by the following algorithm.

Input:  $A$

If  $A$  does not have properties (1)–(3) then reject.

For all induced substructures  $W$  of  $A$ :

// (can be implemented non-deterministically)

If  $W$  is the  $\tau$ -structure of a word  $w \in \Sigma^*$  then

If  $w \notin L$  then reject.

Accept.

Lemma 2 shows that  $A$  homomorphically maps to  $\Gamma$  if and only if  $A$  satisfies (1)–(4), and this is what the algorithm tests. The algorithm can be implemented on a non-deterministic polynomial-time Turing machine such that there exists a run on input  $A$  where the algorithm rejects if and only if  $A$  homomorphically maps to  $\Gamma$ .

## 4 coNP-Intermediate $\omega$ -Categorical Templates

In this section we construct an  $\omega$ -categorical directed graph  $\Gamma$  such that  $\text{CSP}(\Gamma)$  is in coNP, but neither coNP-complete nor in P (unless  $\text{coNP}=\text{P}$ ). As in the previous section, the infinite structures studied here are all defined as Fraïssé limits. All structures in this section will be directed graphs.

Henson [11] used Fraïssé limits to construct  $2^\omega$  many  $\omega$ -categorical directed graphs. If  $\mathcal{N}$  is a class of  $\tau$ -structures,  $\text{Forb}(\mathcal{N})$  denotes the class of all finite  $\tau$ -structures  $A$  such that no structure from  $\mathcal{N}$  embeds into  $A$ . A *tournament* is a directed graph  $G$  (without self-loops) such that for all pairs  $x, y$  of distinct vertices exactly one of the pairs  $(x, y), (y, x)$  is an arc in  $G$ . Note that for all classes  $\mathcal{N}$  of finite tournaments,  $\text{Forb}(\mathcal{N})$  is an amalgamation class, because if  $G_1$  and  $G_2$  are directed graphs in  $\text{Forb}(\mathcal{N})$  such that  $H = G_1 \cap G_2$  is an induced substructure of both  $G_1$  and  $G_2$ , then the free amalgam  $G_1 \cup G_2$  is also in  $\text{Forb}(\mathcal{N})$ . We write  $\Gamma_{\mathcal{N}}$  for the Fraïssé-limit of  $\text{Forb}(\mathcal{N})$ . Observe that for finite  $\mathcal{N}$  the problem  $\text{CSP}(\Gamma_{\mathcal{N}})$  can be solved in deterministic polynomial time, because for a given instance  $S$  of this problem an algorithm simply has to check whether there is a homomorphism from one of the structures in  $\mathcal{N}$  to  $S$ , which is the case if and only if there is a homomorphism from  $S$  to  $\Gamma_{\mathcal{N}}$ .

Henson in his proof specified an infinite set  $\mathcal{T}$  of tournaments  $T_1, T_2, \dots$  with the property that  $T_i$  does not embed into  $T_j$  if  $i \neq j$ . Note that this implies that for two distinct subsets  $\mathcal{T}_1$  and  $\mathcal{T}_2$  of  $\mathcal{T}$  the two sets  $\text{Forb}(\mathcal{T}_1)$  and  $\text{Forb}(\mathcal{T}_2)$  are distinct as well. Since there are  $2^\omega$  many subsets of the infinite set  $\mathcal{T}$ , there are also that many distinct  $\omega$ -categorical directed graphs. The tournament  $T_n$  in Henson’s set  $\mathcal{T}$  has vertices  $0, \dots, n + 1$ , and the following edges:



- $(i, j)$  for  $j = i + 1$  and  $0 \leq i \leq n$ ;
- $(0, n + 1)$ ;
- $(j, i)$  for  $j > i + 1$  and  $(i, j) \neq (0, n + 1)$ .

**Proposition 1.** *The problem  $CSP(\Gamma_{\mathcal{T}})$  is coNP-complete.*

*Proof.* The problem is contained in coNP, because we can efficiently test whether a sequence  $v_1, \dots, v_k$  of distinct vertices of a given directed graph  $G$  induces  $T_k$  in  $G$ , i.e., whether  $(v_i, v_j)$  is an arc in  $G$  if and only if  $(i, j)$  is an arc in  $T_k$ , for all  $i, j \in \{1, \dots, k\}$ . If for all such sequences of vertices this test is negative, we can be sure that  $G$  is from  $Forb(\mathcal{T})$ , and hence homomorphically maps to  $\Gamma_{\mathcal{T}}$ . Otherwise,  $G$  embeds a structure from  $\mathcal{T}$ , and hence does not homomorphically map to  $\Gamma_{\mathcal{T}}$ .

The proof of coNP-hardness goes by reduction from the complement of the NP-complete 3-SAT problem, and is inspired by a classical reduction from 3-SAT to Clique. For a given 3-SAT instance, we create an instance  $G$  of  $CSP(\Gamma_{\mathcal{T}})$  as follows: If

$$\{x_0^1, x_0^2, x_0^3\}, \dots, \{x_{k+1}^1, x_{k+1}^2, x_{k+1}^3\}$$

are the clauses of the 3-SAT formula (we assume without loss of generality that the 3-SAT instance has at least three clauses), then the vertex set of  $G$  is  $\{(0, 1), (0, 2), (0, 3), \dots, (k + 1, 1), (k + 1, 2), (k + 1, 3)\}$ , and the arc set of  $G$  consists of all pairs  $((i, j), (p, q))$  of vertices such that  $x_i^j \neq \neg x_p^q$  and such that  $(i, p)$  is an arc in  $T_k$ .

We claim that a 3-SAT instance is unsatisfiable if and only if the created instance  $G$  homomorphically maps to  $\Gamma_{\mathcal{T}}$ . The 3-SAT instance is satisfiable iff there is a mapping from the variables to true and false such that in each clause at least one literal, say  $x_0^{j_0}, \dots, x_{k+1}^{j_{k+1}}$ , is true. This is the case if and only if the vertices  $(0, j_1), \dots, (k + 1, j_{k+1})$  induce  $T_k$  in  $G$ , i.e.,  $((i, j_i), (p, j_p))$  is an edge if and only if  $(i, p)$  is an edge in  $T_k$ . This is the case if and only if  $T_k$  embeds into  $G$ , and if and only if  $G$  does not homomorphically map to  $\Gamma_{\mathcal{T}}$ .  $\square$

We now modify the proof of Ladner’s Theorem given in [16] (which is basically Ladner’s original proof) to create a subset  $\mathcal{T}_0$  of  $\mathcal{T}$  such that  $CSP(\Gamma_{\mathcal{T}_0})$  is in coNP, but neither in P nor coNP-complete (unless coNP=P). One of the ideas in Ladner’s proof is to ‘blow holes into SAT’, such that the resulting problem is too sparse to be NP-complete and too dense to be in P. Our modification is that we do not blow holes into a computational problem itself, but that we ‘blow holes into the obstruction set  $\mathcal{T}$  of  $CSP(\Gamma_{\mathcal{T}})$ ’.

In the following, we fix one of the standard encodings of graphs as strings over the alphabet  $\{0, 1\}$ . Let  $M_1, M_2, \dots$  be an enumeration of all polynomial-time bounded Turing machines, and let  $R_1, R_2, \dots$  be an enumeration of all polynomial time bounded reductions. We assume that these enumerations are effective; it is well-known that such enumerations exist.

The definition of  $\mathcal{T}_0$  uses a Turing machine  $F$  that computes a function  $f : \mathbb{N} \rightarrow \mathbb{N}$ , which is defined below. The set  $\mathcal{T}_0$  is then defined as follows.

$$\mathcal{T}_0 = \{T_n \mid f(n) \text{ is even} \}$$



The input number  $n$  is given to the machine  $F$  in unary representation. The computation of  $F$  proceeds in two phases. In the first phase,  $F$  simulates itself<sup>2</sup> on input 1, then on input 2, 3, and so on, until the number of computation steps of  $F$  in this phase exceeds  $n$  (we can always maintain a counter during the simulation to recognize when to stop). Let  $k$  be the value  $f(i)$  for the last input  $i$  for which the simulation was completely performed by  $F$ .

In the second phase, the machine stops if phase two takes more than  $n$  computation steps, and  $F$  returns  $k$ . We distinguish whether  $k$  is even or odd. If  $k$  is even, all directed graphs  $G$  on  $s = 1, 2, 3, \dots$  vertices are enumerated. For each directed graph  $G$  in the enumeration the machine  $F$  simulates  $M_{k/2}$  on the encoding of  $G$ . Moreover,  $F$  computes whether  $G$  homomorphically maps to  $\Gamma_{\mathcal{T}_0}$ . This is the case if for all structures  $T_l \in \mathcal{T}$  that embed into  $G$  the value of  $f(l)$  is even. So  $F$  tests for  $l = 1, 2, \dots, s$  whether  $T_l$  embeds to  $G$  ( $F$  uses any straightforward exponential time algorithm for this purpose), and if it does, simulates itself on input  $l$  to find out whether  $f(l)$  is even. If

- (1)  $M_{k/2}$  rejects and  $G$  homomorphically maps to  $\Gamma_{\mathcal{T}_0}$ , or
- (2)  $M_{k/2}$  accepts and  $G$  does not homomorphically map to  $\Gamma_{\mathcal{T}_0}$ ,

then  $F$  returns  $k + 1$  (and  $f(n) = k + 1$ ).

The other case of the second phase is that  $k$  is odd. Again  $F$  enumerates all directed graphs  $G$  on  $s = 1, 2, 3, \dots$  vertices, and simulates the computation of  $R_{\lfloor k/2 \rfloor}$  on the encoding of  $G$ . Then  $F$  computes whether the output of  $R_{\lfloor k/2 \rfloor}$  encodes a directed graph  $G'$  that homomorphically maps to  $\Gamma_{\mathcal{T}_0}$ . The graph  $G'$  homomorphically maps to  $\Gamma_{\mathcal{T}_0}$  iff for all tournaments  $T_l$  that embed into  $G'$  the value  $f(l)$  is even. Whether  $T_l$  embeds into  $G'$  is tested with a straightforward exponential-time algorithm. To test whether  $f(l)$  is even,  $F$  simulates itself on input  $l$ . Finally,  $F$  tests with a straightforward exponential-time algorithm whether  $G$  homomorphically maps to  $\Gamma_{\mathcal{T}}$ . If

- (3)  $G$  homomorphically maps to  $\Gamma_{\mathcal{T}}$  and  $G'$  does not homomorphically map to  $\Gamma_{\mathcal{T}_0}$ , or
- (4)  $G$  does not homomorphically map to  $\Gamma_{\mathcal{T}}$  and  $G'$  homomorphically maps to  $\Gamma_{\mathcal{T}_0}$ ,

then  $F$  returns  $k + 1$ .

**Lemma 3.** *The function  $f$  is a non-decreasing function, that is, for all  $n$  we have  $f(n) \leq f(n + 1)$ .*

**Lemma 4.** *For all  $n_0$  there is  $n > n_0$  such that  $f(n) > f(n_0)$  (unless  $\text{coNP} \neq P$ ).*

*Proof.* Assume for contradiction that there exists an  $n_0$  such that  $f(n)$  equals a constant  $k_0$  for all  $n \geq n_0$ . Then there also exists an  $n_1$  such that for all  $n \geq n_1$  the value of  $k$  computed by the first phase of  $F$  on input  $n$  is  $k_0$ .

---

<sup>2</sup> Note that by the fixpoint theorem of recursion theory we can assume that  $F$  has access to its own description.

If  $k_0$  is even, then on all inputs  $n \geq n_1$  the second phase of  $F$  simulates  $M_{k_0/2}$  on encodings of an enumeration of graphs. Since the output of  $F$  must be  $k_0$ , for all graphs neither (1) nor (2) can apply. Since this holds for all  $n \geq n_1$ , the polynomial-time bounded machine  $M_{k_0/2}$  correctly decides  $\text{CSP}(\Gamma_{\mathcal{T}_0})$ , and hence  $\text{CSP}(\Gamma_{\mathcal{T}_0})$  is in P. But then there is the following polynomial-time algorithm that solves  $\text{CSP}(\Gamma_{\mathcal{T}})$ , a contradiction to coNP-completeness of  $\text{CSP}(\Gamma_{\mathcal{T}})$  (Proposition 1) and our assumption that  $\text{coNP} \neq \text{P}$ .

Input: A directed graph  $G$ .

Test whether  $G$  homomorphically maps to  $\Gamma_{\mathcal{T}_0}$ .

If yes, accept.

If no, test whether one of the finitely many graphs in  $\mathcal{T} \setminus \mathcal{T}_0$  embeds into  $G$ .

Accept if none of them embeds into  $G$ .

Reject otherwise.

If  $k_0$  is odd, then on all inputs  $n \geq n_1$  the second phase of  $F$  does not find a graph  $G$  for which (3) or (4) applies, because the output of  $F$  must be  $k_0$ . Hence,  $R_{\lfloor k_0/2 \rfloor}$  is a polynomial-time reduction from  $\text{CSP}(\Gamma_{\mathcal{T}})$  to  $\text{CSP}(\Gamma_{\mathcal{T}_0})$ , and by Proposition 1 the problem  $\text{CSP}(\Gamma_{\mathcal{T}_0})$  is coNP-hard. But note that because  $f(n)$  equals the odd number  $k_0$  for all but finitely many  $n$ , the set  $\mathcal{T}_0$  is finite. Therefore,  $\text{CSP}(\Gamma_{\mathcal{T}_0})$  can be solved in polynomial time, contradicting our assumption that  $\text{coNP} \neq \text{P}$ .  $\square$

**Theorem 4.**  *$\text{CSP}(\Gamma_{\mathcal{T}_0})$  is in coNP, but neither in P nor coNP-complete (unless  $\text{coNP}=\text{P}$ ).*

*Proof.* It is easy to see that  $\text{CSP}(\Gamma_{\mathcal{T}_0})$  is in coNP. On input  $G$  the algorithm non-deterministically chooses a sequence of  $l$  vertices, and checks in polynomial time whether this sequence induces a copy of  $T_l$ . If yes, the algorithm computes  $f(l)$ , which can be done in linear time by executing  $F$  on the unary representation of  $l$ . If  $f(l)$  is even, the algorithm accepts. Recall that  $G$  does not homomorphically map to  $\Gamma_{\mathcal{T}_0}$  iff a tournament  $T_l \in \mathcal{T}_0$  embeds into  $G$ , which is the case iff there is an accepting computation path for the above non-deterministic algorithm.

Suppose that  $\text{CSP}(\Gamma_{\mathcal{T}_0})$  is in P. Then for some  $i$  the machine  $M_i$  decides  $\text{CSP}(\Gamma_{\mathcal{T}_0})$ . By Lemma 3 and Lemma 4 there exists an  $n_0$  such that  $f(n_0) = 2i$ . Then there must also be an  $n_1 > n_2$  such that the value  $k$  computed during the first phase of  $F$  on input  $n_1$  equals  $2i$ . Since  $M_i$  correctly decides  $\text{CSP}(\Gamma_{\mathcal{T}_0})$ , the machine  $F$  returns  $2i$  on input  $n_1$ . By Lemma 3, the machine  $F$  also returns  $2i$  for all inputs from  $n_1$  to  $n_2$ , and by induction it follows that it  $F$  returns  $2i$  for all inputs larger than  $n \geq n_0$ , in contradiction to Lemma 4.

Finally, suppose that  $\text{CSP}(\Gamma_{\mathcal{T}_0})$  is coNP-complete. Then for some  $i$  the machine  $R_i$  is a valid reduction from  $\text{CSP}(\Gamma_{\mathcal{T}})$  to  $\text{CSP}(\Gamma_{\mathcal{T}_0})$ . Again, by Lemma 3 and Lemma 4 there exists an  $n_1$  such that the value  $k$  computed during the first phase of  $F$  on input  $n_1$  equals  $2i$ . Since the reduction  $R_i$  is correct, the machine  $F$  returns  $2i$  on input  $n_1$ , and in fact returns  $2i$  on all inputs greater than  $n_1$ . This contradicts Lemma 4.  $\square$

## 5 Left-Hand Side Restrictions

Let  $\mathcal{S}$  be a class of finite  $\tau$ -structures. Then  $\text{CSP}(\mathcal{S}, \_)$  is the computational problem to decide whether for a given pair  $(A, B)$  of finite  $\tau$ -structures with  $A \in \mathcal{S}$  there is a homomorphism from  $A$  to  $B$ .

As an example, let  $\tau$  be the signature that consists of a single binary relation. In this case,  $\tau$ -structures can be considered as directed graphs (potentially with loops). If  $\mathcal{C}$  is the set of all complete graphs (without loops!), then  $\text{CSP}(\mathcal{C}, \_)$  is essentially a formulation of the Clique problem.

The following question was left open in [10]: *Are there classes of structures  $\mathcal{S}$  such that  $\text{CSP}(\mathcal{S}, \_)$  is in NP, but neither in P nor NP-complete?*

We answer this question positively, and construct such a class  $\mathcal{S}$ , which can even be decided in polynomial time. Again we use a modification of Ladner's theorem. The modification is similar to the modification presented in Section 4. This time, we 'blow holes into the possible clique sizes for the clique problem' and obtain a class  $\mathcal{C}_0 \subseteq \mathcal{C}$  such that  $\text{CSP}(\mathcal{C}_0, \_)$  is in  $\text{NP} \setminus \text{P}$  and not NP-complete (unless  $\text{P}=\text{NP}$ ).

The idea is to define  $\mathcal{C}_0$  in such a way that the  $\mathcal{C} \setminus \mathcal{C}_0$  becomes finite when  $\text{CSP}(\mathcal{C}_0, \_)$  is in P; hence,  $\text{CSP}(\mathcal{C}_0, \_)$  is polynomial-time equivalent to the Clique problem, a contradiction unless  $\text{P}=\text{NP}$ . Moreover, the construction of  $\mathcal{C}_0$  is such that  $\mathcal{C}_0$  is finite if  $\text{CSP}(\mathcal{C}_0, \_)$  is NP-hard. But for finite  $\mathcal{C}_0$ , the problem  $\text{CSP}(\mathcal{C}_0, \_)$  is in P, again contradicting the assumption that  $\text{P} \neq \text{NP}$ . We also take extra care to make  $\mathcal{C}_0$  polynomial-time decidable.

**Theorem 5.**  *$\text{CSP}(\mathcal{C}_0, \_)$  is in NP, but neither in P nor NP-complete (unless  $\text{P}=\text{NP}$ ). Moreover, the set  $\mathcal{C}_0$  can be decided in deterministic polynomial time.*

In our proof we do not use any specific properties of the class  $\mathcal{C}$  and the clique problem, but in fact we can construct classes  $\mathcal{S}_0 \subseteq \mathcal{S}$  with NP-intermediate  $\text{CSP}(\mathcal{S}_0, \_)$  for any class  $\mathcal{S}$  where  $\text{CSP}(\mathcal{S}, \_)$  is NP-complete.

## 6 The Local-Global Conjecture

The complexity of the constraint satisfaction problem has also been studied for templates  $\Gamma$  with an *infinite signature*. Several well-known computational problems can be modeled as CSPs only if we allow (countably) infinite constraint languages: examples are boolean Horn-satisfiability, Ord-Horn constraints in temporal reasoning [15], or solving linear equation systems over a finite field.

If the local-global conjecture as stated in Section 1 is true, then the dichotomy for finite constraint languages implies the dichotomy for infinite constraint languages: if an infinite constraint language  $\Gamma$  has a finite reduct  $\Gamma'$  such that  $\text{CSP}(\Gamma')$  is NP-hard, then  $\text{CSP}(\Gamma)$  is clearly NP-hard as well. On the other hand, if  $\text{CSP}(\Gamma)$  is locally tractable, then the conjecture implies that  $\text{CSP}(\Gamma)$  is globally tractable.

We show that if the local-global conjecture is false, then there is no dichotomy for infinite constraint languages.

**Theorem 6.** *If the local-global conjecture is false, then there exists a template  $\Gamma_0$  with finite domain  $D$  and infinite signature such that  $\text{CSP}(\Gamma_0)$  is neither globally tractable nor NP-complete (unless  $P=NP$ ); moreover, the meta-problem for  $\Gamma_0$  is efficiently decidable, i.e., given a relation  $R$  over  $D$ , we can decide in polynomial-time whether  $R$  is in  $\Gamma_0$ .*

The proof of Theorem 6 is essentially again a modification of Ladner’s theorem, but we have to overcome a complication: for a straightforward application of Ladner’s theorem, we need that if  $\Gamma$  is an expansion of  $\Gamma'$  by finitely many relation symbols, and  $\text{CSP}(\Gamma)$  is NP-hard, then  $\text{CSP}(\Gamma')$  is NP-hard as well: but this is not true in general. We only sketch the basic setting of Ladner’s construction, which we have already seen twice in this paper, and focus on the complication.

*Proof (Sketch).* Assume that there is an infinite constraint language  $\Gamma$  with a finite domain such that  $\text{CSP}(\Gamma)$  is locally tractable, but not globally tractable. If  $\Gamma$  is not NP-complete, we are already done with  $\Gamma_0 = \Gamma$ , so assume that  $\Gamma$  is NP-complete. We claim that we can assume without loss of generality that  $\Gamma$  contains all primitive positive definable relations (for an introduction to this basic concept in model theory and its applications in constraint satisfaction theory, see e.g. [6]). To see this, first observe that the expansion  $\Gamma'$  of  $\Gamma$  by all those relations trivially still has an NP-complete CSP. Moreover, all reducts of  $\Gamma'$  with a finite signature have a CSP that is in  $P$ , because all relations in this reduct can be defined by finitely many relations from  $\Gamma$ . So,  $\Gamma'$  can be obtained from a finite reduct of  $\Gamma$  by expansion with finitely many primitive positive relations, and hence  $\text{CSP}(\Gamma')$  is in  $P$ .

We construct a reduct  $\Gamma_0$  of  $\Gamma$  such that  $\text{CSP}(\Gamma_0)$  is neither in  $P$  nor NP-complete (unless  $P=NP$ ). Again, the definition of  $\Gamma_0$  is by a Turing machine  $F$  that computes a function  $f : \mathbb{N} \rightarrow \mathbb{N}$ , and the  $n$ -th relation of  $\Gamma$  (according to some fixed enumeration of the relations of  $\Gamma$ ) is in  $\Gamma_0$  if  $f(n)$  is even.

As in the proofs before,  $n$  is given to the machine  $F$  in unary representation, and we can define  $F$  in such a way that

- it runs in polynomial time in  $n$ , and
- $\Gamma_0$  becomes finite if  $\text{CSP}(\Gamma_0)$  is NP-hard, and
- $\Gamma$  is an expansion of  $\Gamma_0$  by finitely many relations if  $\text{CSP}(\Gamma_0)$  is in  $P$ .

If  $\Gamma_0$  is finite, then  $\text{CSP}(\Gamma_0)$  is tractable, because every reduct of  $\Gamma$  with a finite signature is by assumption tractable, and we obtain that  $P=NP$ . Now, suppose that we are in the other case, and that  $\Gamma$  is an expansion of  $\Gamma_0$  by finitely many relations. We want to show that  $\text{CSP}(\Gamma_0)$  is NP-hard by reducing  $\text{CSP}(\Gamma)$  to  $\text{CSP}(\Gamma_0)$ .

Let  $S$  be an instance of  $\text{CSP}(\Gamma)$ . Note that  $S$  might contain constraints for the relations from  $\Gamma$  that are not in  $\Gamma_0$ , but there is a  $k$  such that all those constraints have arity less than  $k$ . Because  $\Gamma$  contains all primitive positive definable relations,  $\Gamma$  contains in particular for every  $l$ -ary relation  $R$  the  $k$ -ary relation  $R' := R \times D \times \dots \times D$ . We now replace each constraint in  $S$  with a relation

$R$  from  $\Gamma$  that is not in  $\Gamma_0$  by a constraint for  $R'$ , introducing  $k - l$  new dummy variables for the last  $k - l$  arguments of  $R'$ . Even though the representation of  $R'$  is much larger than the representation of  $R$ , this can only lead to a linear increase in the size of the instance  $S$ , because both  $D$  and  $k$  are fixed. The resulting structure  $S'$  is an instance of  $\text{CSP}(\Gamma_0)$ , and  $S'$  homomorphically maps to  $\Gamma_0$  if and only if  $S$  homomorphically maps to  $\Gamma$ . Therefore  $\text{CSP}(\Gamma_0)$  is NP-complete, which again implies that  $\text{P}=\text{NP}$ .  $\square$

*Acknowledgements.* We would like to thank Hubie Chen for his helpful comments on an earlier version of this paper.

## References

1. Bauslaugh, B.L.: The complexity of infinite h-coloring. *J. Comb. Theory, Ser. B* 61(2), 141–154 (1994)
2. Bodirsky, M.: Constraint satisfaction problems with infinite templates. Survey (to appear, 2007)
3. Bulatov, A.: Tractable conservative constraint satisfaction problems. In: Proceedings of LICS 2003, pp. 321–330 (2003)
4. Bulatov, A.: A graph of a relational structure and constraint satisfaction problems. In: Proceedings of LICS 2004, Turku, Finland (2004)
5. Bulatov, A., Jeavons, P.: Algebraic structures in combinatorial problems. Technical report MATH-AL-4-2001, Technische Universitat Dresden, submitted to International Journal of Algebra and Computing (2001)
6. Bulatov, A., Krokhin, A., Jeavons, P.G.: Classifying the complexity of constraints using finite algebras. *SIAM Journal on Computing* 34, 720–742 (2005)
7. Chen, Y., Thurley, M., Weyer, M.: Understanding the complexity of induced substructure isomorphisms. In: Aceto, L., Damgaard, I., Goldberg, L.A., Halldorsson, M.M., Ingolfsson, A., Walukiewicz, I. (eds.) ICALP 2008. LNCS, vol. 5125. Springer, Heidelberg (2008)
8. Cherlin, G.: The classification of countable homogeneous directed graphs and countable homogeneous  $n$ -tournaments. *AMS Memoir.* 131(621) (January 1998)
9. Feder, T., Vardi, M.: The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory. *SIAM Journal on Computing* 28, 57–104 (1999)
10. Grohe, M.: The complexity of homomorphism and constraint satisfaction problems seen from the other side. *Journal of the ACM* 54(1) (2007)
11. Henson, C.W.: Countable homogeneous relational systems and categorical theories. *Journal of Symbolic Logic* 37, 494–500 (1972)
12. Hodges, W.: A shorter model theory. Cambridge University Press, Cambridge (1997)
13. Idziak, P.M., Markovic, P., McKenzie, R., Valeriote, M., Willard, R.: Tractability and learnability arising from algebras with few subpowers. In: LICS 2007, pp. 213–224 (2007)
14. Ladner, R.E.: On the structure of polynomial time reducibility. *J. ACM* 22(1), 155–171 (1975)
15. Nebel, B., Bürckert, H.-J.: Reasoning about temporal relations: A maximal tractable subclass of Allen’s interval algebra. *J. ACM* 42(1), 43–66 (1995)
16. Papadimitriou, C.H.: Computational Complexity. Addison-Wesley, Reading (1994)
17. Schwandtner, G.: Datalog on infinite structures. Dissertation, Humboldt-Universität zu Berlin (submitted, 2008)