Locally Finite Constraint Satisfaction Problems

Bartek Klin*, Eryk Kopczyński†, Joanna Ochremiak*, Szymon Toruńczyk*
University of Warsaw
Email: {klin,erykk,ochremiak,szymtor}@mimuw.edu.pl

***Abstract*—First-order definable structures with atoms are infinite, but exhibit enough symmetry to be effectively manipulated. We study Constraint Satisfaction Problems (CSPs) where both the instance and the template are definable structures with atoms. As an initial step, we consider locally finite templates, which contain potentially infinitely many finite relations. We argue that such templates occur naturally in Descriptive Complexity Theory.**

We study CSPs over such templates for both finite and infinite, definable instances. In the latter case even decidability is not obvious, and to prove it we apply results from topological dynamics. For finite instances, we show that some central results from the classical algebraic theory of CSPs still hold: the complexity is determined by polymorphisms of the template, and the existence of certain polymorphisms, such as majority or Maltsev polymorphisms, guarantees the correctness of classical algorithms for solving finite CSP instances.

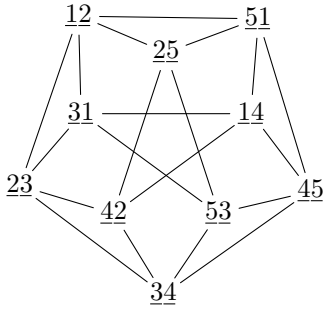
***Index Terms*—Sets with atoms; Constraint Satisfaction Problems**

Once and for all, fix a countably infinite set $\mathcal{A} = \{\underline{1}, \underline{2}, \underline{3}, \ldots\}$, whose elements we call *atoms*.

I. Introduction

Example 1. Here is an easy puzzle: consider an (infinite) graph G with ordered pairs of distinct atoms as vertices (here we denote such a pair simply by ab, for $a \neq b \in \mathcal{A}$), and with an undirected edge ab—bc whenever a and c are distinct. Is this graph 3-colorable?

The answer is negative, as G contains the subgraph:



which, as can be checked by hand, is not 3-colorable. We do not know whether this is the smallest non-3-colorable subgraph of G, but we could not find a smaller one, and it is rather interesting to see how big a graph we needed to check.

This motivates a harder puzzle: consider graphs with n-tuples of distinct atoms as vertices, and edges defined by quantifier-free (or even first order) formulas with equality, and with $2n$ variables that range over atoms; for G above, $n = 2$ and the set of edges is:

$$\big\{\{ab, dc\} \mid a, b, c, d \in \mathcal{A}, (b = d) \wedge (a \neq b) \wedge (c \neq d) \wedge (a \neq c)\big\}.$$

Now the question is whether the 3-colorability of a graph represented by a number n and a formula is decidable at all? It is a standard exercise in logical compactness that a graph is 3-colorable iff all its finite subgraphs are, but it may not be clear whether there is a computable bound on the size of finite subgraphs that need to be checked to ensure the colorability of the entire graph. □

Example 2. Systems of linear equations over the two-element field $\mathbb{Z}_2$ can be augmented with atoms just as graphs can. Consider n-tuples of distinct atoms as variable names, and let a system of equations be defined by a formula similarly to Example 1, for example (with $n = 2$):

$$\big\{ab + bc + ca = 0 \mid a, b, c \in \mathcal{A}, (a \neq b) \wedge (b \neq c) \wedge (a \neq c)\big\}.$$

This system has a trivial solution where all variables have value 0. To disallow that solution one may e.g. extend the system by one more equation:

$$\underline{12} + \underline{21} = 1.$$

Does the extended system have a solution? It turns out that it does not, but a finite subsystem with no solutions again turns out curiously bulky. Here is the smallest one that we have managed to find:

$$\begin{array}{cccccccccccccc}
\underline{12} & + & \underline{21} & & & & & & & & & & & = 1 \\
\underline{12} & & + & \underline{23} & + & \underline{31} & & & & & & & & = 0 \\
 & \underline{21} & & & + & \underline{13} & + & \underline{32} & & & & & & = 0 \\
 & & \underline{23} & & & & + & \underline{34} & + & \underline{42} & & & & = 0 \\
 & & & \underline{31} & & & & & + & \underline{15} & & + & \underline{53} & = 0 \\
 & & & & \underline{13} & & + & \underline{34} & & & & & + & \underline{41} = 0 \\
 & & & & & \underline{32} & & & & & + & \underline{25} & + & \underline{53} = 0 \\
 & & & & & & & \underline{42} & & + & \underline{25} & & + & \underline{54} = 0 \\
 & & & & & & & & \underline{15} & & & & + & \underline{54} + \underline{41} = 0 \\
\hline
0 & + & 0 & + & 0 & + & 0 & + & 0 & + & 0 & + & 0 & = 1
\end{array}$$

Again, a question appears whether the solvability of equation systems given by formulas over atoms is decidable, and if so, what its complexity is. □

The above are examples of so-called *Constraint Satisfaction Problems* (CSPs). An instance $\mathbb{I}$ of a CSP is a set of *variables* together with a set of *constraints* of the form

Supported by the Polish National Science Centre (NCN) grants *2012/07/B/ST6/01497 and †2012/07/D/ST6/02435.

$\big((x_1,\dots,x_n),R\big)$, where the x_i are variables and R is an n-ary relation belonging to a fixed family of relations $\mathcal{R}$ over a *domain* T; the pair $\mathbb{T}=(T,\mathcal{R})$ is called a *template* for $\mathbb{I}$.

For example, 3-colorability is a CSP for a template with three elements (colors) equipped with a single binary inequality relation $\neq$. To see a graph as an instance, one considers its vertices as variables, and adds a constraint $\big((x,y),\neq\big)$ whenever x and y are adjacent. An equation system $\mathbb{E}$ over $\mathbb{Z}_2$, assuming that every equation is of the form $x+y+z=0$ or $x+y=1$, can be seen as an instance over a template with two elements 0 and 1, equipped with two relations:

$$\begin{array}{rcl} Z & = & \{(0,0,0),(0,1,1),(1,0,1),(1,1,0)\},\\ S & = & \{(0,1),(1,0)\}. \end{array}$$

To construct an instance, one picks constraints:

$$\begin{array}{rl} \big((x,y,z),Z\big) & \text{for each} \quad x+y+z=0 \text{ in } \mathbb{E},\\ \big((x,y),S\big) & \text{for each} \quad x+y=1 \text{ in } \mathbb{E}. \end{array}$$

A *solution* of an instance is an assignment f which maps every variable to a template element, so that for every constraint $\big((x_1,\dots,x_n),R\big)$, the tuple of values $\big(f(x_1,),\dots,f(x_n)\big)$ belongs to the relation R. It is useful to view a template $\mathbb{T}=(T,\mathcal{R})$ as a relational structure with universe[1] T, over the signature $\mathcal{R}$, with the tautological interpretation mapping. A CSP instance $\mathbb{I}$ over the template $\mathbb{T}$ can then be viewed as a relational structure, whose universe consists of its variables I, and the interpretation of a relation $R\in\mathcal{R}$ of arity n is the set of those tuples $\bar{x}\in I^n$ for which $(\bar{x},R)$ is a constraint. Then solutions of $\mathbb{I}$ correspond to homomorphisms of relational structures from $\mathbb{I}$ to $\mathbb{T}$.

The classical theory of CSPs tries to classify the computational complexity of the following decision problem, parametrized by a template $\mathbb{T}$, with *finite* instances.

Problem: CSP($\mathbb{T}$)
Input: A finite instance $\mathbb{I}$ over $\mathbb{T}$
Decide: Does $\mathbb{I}$ have a solution?

The *algebraic approach* to this end is particularly successful. It is based on the observation that the complexity of CSP($\mathbb{T}$) entirely depends on the algebra of *polymorphisms* (a multivariate generalization of the notion of an endomorphism) of the template $\mathbb{T}$ [1]. For example, the fact that finite systems of equations over $\mathbb{Z}_2$ can be solved in polynomial time can be inferred from the fact that the relevant template has a so-called Maltsev polymorphism [2], and the NP-completeness of graph 3-coloring follows from the fact that the corresponding template has no so-called cyclic polymorphisms [3]–[5].

Our Examples 1 and 2 do not fit the mainstream development of CSP theory, since our instances are infinite. They are, however, definable via first order expressions, in a sense made precise in Section II. The aim of this paper is to formulate the rudiments of CSP theory for definable structures. We define and study the complexity of the following decision problem.

[1]In this paper, we adapt the convention that the universe of a relational structure $\mathbb{A}$ is denoted with the corresponding italic letter A.

Problem: CSP-Inf($\mathbb{T}$)
Input: An expression defining an instance $\mathbb{I}$ over $\mathbb{T}$
Decide: Does $\mathbb{I}$ have a solution?

We show that, for any fixed finite template $\mathbb{T}$, this problem is decidable, and specify tight complexity bounds. In particular, the following result is a consequence of the results proved in Section III.

Theorem 3. *Let $\mathbb{T}$ be a finite template such that* CSP($\mathbb{T}$) *is complete for a complexity class $\mathcal{C}$ under logarithmic space reductions. Then* CSP-Inf($\mathbb{T}$) *is decidable and complete for the complexity class* $\exp(\mathcal{C})$ *under logarithmic space reductions.*

The general definition of the class $\exp(\mathcal{C})$ is given in Section III-B; here we just mention that $\exp(\mathrm{L}) = \textsc{PSpace}, \exp(\mathrm{P}) = \textsc{Exp}, \exp(\mathrm{NP}) = \textsc{NExp}$, etc.

Interestingly, our key technical tool for proving the upper bound comes from topological dynamics, in the following theorem due to Pestov:

Theorem 4 ([6])**.** *Every continuous action of the topological group* $\mathrm{Aut}(\mathbb{Q},\leq)$ *on a compact space has a fixpoint.*

This theorem is strongly related with Ramsey's theorem (see [7] for a generalization of this theorem, linking it to Ramsey theory). In fact, the upper bound in Theorem 3 could be proved directly with the use of Ramsey's theorem.

In Section IV, we reverse the situation and consider finite instances over infinite templates. We allow the templates to have an infinite set of relations, but we assume them to be *locally finite*, i.e., every relation is finite. Examples of such CSPs appear in various contexts:

Example 5. Consider the following *graph coloring* problem. Fix a positive integer k. Let $G=(V,E)$ be a finite graph, together with a labeling $l: V\to\binom{\mathcal{A}}{k}$, where elements of $\mathcal{A}$ are interpreted as *colors*.

The problem is to decide whether one can find a coloring $c: V\to\mathcal{A}$ such that $c(v)\in l(v)$ for every $v\in V$ and $c(v)\neq c(w)$ whenever v and w are adjacent in G.

This problem can be understood as a CSP over a (definable) template whose domain is $\mathcal{A}$, with a relation

$$R_{C,D}=C\times D-\Delta$$

for every pair $C,D\in\binom{\mathcal{A}}{k}$, where $\Delta\subseteq\mathcal{A}^2$ is the binary diagonal relation. The instance corresponding to a graph has V as the set of variables, and a constraint $\big((v,w),R_{l(v),l(w)}\big)$ for each pair of adjacent vertices v,w. □

Note that while every finite instance over an infinite, locally finite template is trivially an instance over its finite subtemplate, there may be no single finite subtemplate that immediately fits all instances of interest. Since templates in CSP theory are means of grouping large classes of similar instances, it may sometimes be useful to consider infinite, locally finite templates.

Situations where the set of admissible values for variables is not fixed for all instances sometimes arise naturally. For

example, consider the *cycle cover problem*: given a directed graph G, decide whether it contains a set of directed cycles so that every vertex belongs to exactly one cycle. This is equivalent to checking whether one can choose an outgoing edge from every vertex so that no two chosen edges have the same target. In other words, one wants to color every vertex with one of its out-neighbours so that no two vertices get the same color. Assuming a bound k on the out-degree of the vertices of G, and labeling vertices of G with atoms in an arbitrary way, this can be seen as a CSP instance over the template from Example 5. Here graph vertices play the role of colors, therefore the set of possible colors depends on the instance.

Another example is that of Cai-Fürer-Immerman (CFI) graphs [8] considered in Descriptive Complexity Theory:

Example 6. Consider a template with atoms as elements and, for every triple of pairs of distinct atoms $\beta = ((a,a'),(b,b'),(c,c'))$, a ternary relation:

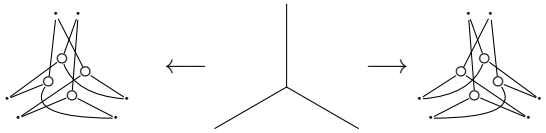
$$R_\beta = \big\{(a,b,c),(a,b',c'),(a',b,c'),(a',b',c)\big\}.$$

To construct an interesting instance over this template, start with a 3-regular graph G and label each edge e with a set $\{a,a'\} \subseteq \mathcal{A}$ of two distinct atoms, so that labels of distinct edges do not intersect. Let the edges of G be the variables of the instance. For each vertex v adjacent to some edges e_1, e_2 and e_3 add exactly one constraint

$$\big((e_1,e_2,e_3),R_\beta\big),$$

where $\beta = \big((a,a'),(b,b'),(c,c')\big)$ arises from some ordering of the unordered labels $\{a,a'\},\{b,b'\},\{c,c'\}$ of the edges e_1,e_2,e_3. Note that even though there are eight possible orderings, there are only two possible resulting relations R_β.

Graphically, this can be seen as replacing every edge with two nodes, and every vertex with one of two little hypergraphs:



Such an instance has a solution if and only if one can choose one of the two nodes for each edge of G so that for each vertex v of G, the three nodes chosen for the adjacent edges are connected by a hyperedge in the hypergraph for v. Checking whether a given instance has a solution is called a *CFI query*, and is the core of the Cai-Fürer-Immerman theorem from Descriptive Complexity (see Section V for more on this topic). □

In Section IV we lift some central results from classical CSP theory to the setting of locally finite templates. First, we prove that the cornerstone result of the algebraic approach to CSP still holds: the complexity of the CSP problem over a template depends only on the set of polymorphisms over that template. We also show that the important notion of a *core* template has the expected properties, and that a definable template can be effectively converted to its core.

Furthermore, we show that a locally finite template admits a family of polymorphisms defined by a specified set of *linear identities* iff each of its finite subtemplates admit such polymorphisms. This proves a series of results analogous to finite CSP theory: for example, if a locally finite template $\mathbb{T}$ has a majority polymorphism or a Maltsev polymorphism then specific polynomial time algorithms correctly solve finite instances over $\mathbb{T}$. Moreover, by the results of Section III, we can effectively decide whether a given definable, locally finite template $\mathbb{T}$ admits polymorphisms which satisfy a specified set of linear identities. This allows us to prove statements such as the one below, which follows from Corollary 35 and Proposition 31 in Section IV, and characterizes those templates for which the CSP problem can be solved by a certain well-known *bounded width algorithm*.

Theorem 7. *Let $\mathbb{T}$ be a definable, locally finite template. It is decidable whether $\mathbb{T}$ has bounded width.*

Coming back to Example 5, one may notice that not only every particular instance there is over a finite template: in fact, the entire graph coloring problem can be presented as a CSP over a single finite template. To do this, impose a total order on $\mathcal{A}$; this defines a bijection from each $l(v)$ to the set $[k] = \{1,2,\dots,k\}$. One can then consider a template of k elements, with a relation

$$R_{i,j} = [k] \times [k] - \{(i,j)\}$$

for each $i,j \in [k]$, and translate a graph to be colored to an instance over that finite template. In Section IV-C, we show how this can be generalized: for any definable, locally finite template $\mathbb{T}$ there is a finite template $\hat{\mathbb{T}}$, and a polynomial time reduction of instances over $\mathbb{T}$ to equivalent instances over $\hat{\mathbb{T}}$. This shows that CSP($\mathbb{T}$) is not computationally harder then CSP($\hat{\mathbb{T}}$). Moreover, if $\hat{\mathbb{T}}$ admits polymorphisms satisfying some linear identities, then so does $\mathbb{T}$. By the algebraic results proved earlier in Section IV, and by results very recently announced by Barto and Pinsker [9], this says that CSP($\hat{\mathbb{T}}$) is not harder than CSP($\mathbb{T}$).

Finally, in Section V we show how locally finite templates streamline the (previously unpleasantly technical) proof of the main result in [10], a characterization of those *linearly patched structures* over which the least fixpoint logic LFP captures polynomial time computations.

Related work

CSP for certain infinite instances were studied in [11]. All instances there are *periodic*, i.e., invariant under an action of a subgroup of finite index of the automorphism group of the total order of integers. This is similar to our approach, as our definable instances are also invariant under certain groups, in particular the automorphism group of rational numbers. However, the choice of a group makes a big difference: thanks to model-theoretic properties of rationals we are able to prove decidability results that do not hold in the setting of [11]. Indeed, one of the main points of that paper was to show

that for periodic instances, 3-colorability is *un*decidable. Proof techniques used there are also quite different from ours.

The line of research started in [11] was continued in [12], and certain infinite instances were studied also in [13]. In [12], [13] it is argued that infinite periodic instances naturally arise when studying large – perhaps of unknown size or infinite – constraint networks whose constraints possess a high degree of regularity or symmetry. We believe that relaxing the periodicity assumption might be natural in many cases, and, as we show, leads to a drastic improvement in the complexity (in the case of 3-colorability, from undecidable by [11] to EXP by Theorem 3). It would be interesting to look for a common generalisation of these developments and ours.

More attention has been devoted to the study of finite instances over infinite templates. In contrast to our work, the templates there usually consist of only finitely many infinite relations. In the most well-behaved case of *ω-categorical* templates, central results of finite CSP still hold [14]–[16]. In particular, the complexity of templates depend only on their polymorphisms, which gives complexity classifications for large classes of ω-categorical templates [17]–[19]. Connections to Ramsey theory were studied in [20]. Our section IV shows that some of these results hold for locally finite templates as well.

Sets with atoms are also known in Computer Science as nominal sets [21]. In fact, our notion of definable set is almost the same as the notion of orbit-finite set considered there: every definable set is orbit-finite, and every orbit-finite nominal set is isomorphic to a definable one [22]. We choose to define our sets by first order formulas, but all results we show here could be reformulated in terms of group actions, orbit-finite sets and finite supports, studied in [21]. In fact, we used that terminology in most previous work on computation theory over sets with atoms [10], [23], [24], of which the present paper is a natural continuation.

II. SETS WITH ATOMS

A. Definable sets

We introduce definable sets with atoms as follows. An *expression* is either a variable ranging over atoms, or a finite tuple of expressions, or a formal finite union of expressions, or an integer, or a *set-builder expression*, which is a variable binding construct of the form

$$\{\ e \mid v_1, \dots, v_n \in \mathcal{A}, \phi\},$$

where e is an expression, $v_1, \dots, v_n$ are bound variables, and ϕ is a first order formula with equality as the only predicate, whose free variables are contained in the free variables of e. A *quantifier-free* expression is an expression which uses only quantifier-free formulas, on every level, recursively.

If an expression e has free variables V, then any valuation $\mathrm{val} : V \to \mathcal{A}$ defines in an obvious way the value $X = e[\mathrm{val}]$, which is either a set, an atom, a tuple, or an integer.[2] We say that X is a *definable set with atoms* and that it is *defined* by e with valuation val. Note that the same set X can be defined by many different expressions. We denote by $\mathcal{D}$ the set of all definable sets. Observe that any element of a definable set is definable.

Example 8. Examples of definable sets with atoms include:

- any atom, such as $\underline{1}$, as defined by the expression v, with valuation $v \mapsto \underline{1}$.
- any pair of atoms, such as $(\underline{1}, \underline{2})$, as defined by the expression (v, w), with valuation $v \mapsto \underline{1}, w \mapsto \underline{2}$.
- the set $\mathcal{A}$ of atoms itself, as defined by the expression $\{v \mid v \in \mathcal{A}\}$.
- for any $n \in \mathbb{N}$, the set $\mathcal{A}^n$ of all n-tuples and $\mathcal{A}^{(n)}$ of non-repeating n-tuples of atoms, as defined by the expression

  $$\{(v_1, \dots, v_n) \mid v_1, \dots, v_n \in \mathcal{A}, \phi\},$$

  where ϕ is $\top$ in the case of $\mathcal{A}^n$ and $\bigwedge_{1 \le i < j \le n} (v_i \neq v_j)$ in the case of $\mathcal{A}^{(n)}$.

An example of a definable function is the swapping function $s : \mathcal{A}^2 \to \mathcal{A}^2$, $s(a, b) = (b, a)$, whose graph is defined by the expression $\{((v, w), (w, v)) \mid v, w \in \mathcal{A}\}$. □

For any mathematical object (a relation, a function, a logical structure, etc.), it makes sense to ask whether it is definable. E.g., a definable relation on X, Y is a relation $R \subseteq X \times Y$ which is a definable set. As a side remark, definable structures over a finite signature correspond, up to isomorphism, to structures which *interpret* in $\mathcal{A}$ (a notion from logic).

As a particular case of the above definition, a *definable instance* is an instance $I = (V, C)$ such that the set of variables V and the set of constraints C are definable. Definable instances are represented by expressions, which are used as inputs for the problem CSP-Inf($\mathbb{T}$) described in the Introduction.

Example 9. We show an expression describing the instance from Example 1. Consider the following expressions.

$$R : \big\{(0,1),(0,2),(1,0),(1,2),(2,0),(2,1)\big\}$$
$$V : \big\{(a,b) \mid a, b \in \mathcal{A}, a \neq b\big\}$$
$$C : \big\{\big(((a,b),(b,c)), R\big) \mid a, b, c \in \mathcal{A}, a \neq b \wedge b \neq c \wedge a \neq c\big\}$$
$$I : (V, C)$$

They define, respectively, the inequality relation on the set of integers $\{0, 1, 2\}$, the variables and the constraints of the instance described in Example 1, and finally the instance itself. In general, a definable instance may use parameters. □

B. Group actions, equivariant sets and orbits

Recall that a group G *acts* on a set U if a mapping $G \times U \to U$ is provided, denoted $(\pi, u) \mapsto \pi \cdot u$, such that $1 \cdot u = u$ and $(\pi \cdot \sigma) \cdot u = \pi \cdot (\sigma \cdot u)$, for all $\pi, \sigma \in G, u \in U$, and 1 the identity element in G. An *orbit* under this action is any set of the form $\{\pi \cdot u : \pi \in G\}$, where $u \in U$.

Let $\mathrm{Aut}(\mathcal{A})$ denote the group of *atom permutations*, i.e., bijections $\pi : \mathcal{A} \to \mathcal{A}$. If $\mathrm{Aut}(\mathcal{A})$ acts on a set U, then we

[2] Tuples and integers could be encoded by standard set-theoretic tricks, but to improve readability we refrain from that. To interpret unions of non-sets we may treat the latter as singleton sets.

say that U is *equivariant*. Note that every equivariant set is the disjoint union of its orbits.

The group $\mathrm{Aut}(\mathcal{A})$ naturally acts on the set $\mathcal{D}$ of definable sets. Indeed, if $\pi \in \mathrm{Aut}(\mathcal{A})$ and X is a set defined by an expression e and valuation val, then let

$$\pi \cdot X = \pi \cdot e[\mathrm{val}] \stackrel{\text{def}}{=} e[\mathrm{val}; \pi],$$

where $\mathrm{val}; \pi$ denotes function composition (i.e., $(\mathrm{val}; \pi)(v) = \pi(\mathrm{val}(v))$). This is well defined: it is easy to prove by induction on expressions that $e[\mathrm{val}] = e'[\mathrm{val}']$ implies $e[\mathrm{val}; \pi] = e'[\mathrm{val}'; \pi]$. For example, if $\pi(\underline{1}) = \underline{2}$ and $\pi(\underline{2}) = \underline{3}$, then $\pi \cdot (\underline{1}, \underline{2}) = (\underline{2}, \underline{3})$ and $\pi \cdot \mathcal{A}^2 = \mathcal{A}^2$.

As a result, the group $\mathrm{Aut}(\mathcal{A})$ acts on $\mathcal{D}$. Moreover, $x \in X$ implies $\pi \cdot x \in \pi \cdot X$, for any $\pi \in \mathrm{Aut}(\mathcal{A})$ and definable sets x, X. We say that a definable set X is equivariant if $\pi \cdot X = X$ for all $\pi \in \mathrm{Aut}(\mathcal{A})$. This is consistent with the previous definition of equivariance, since $\mathrm{Aut}(\mathcal{A})$ then acts on X. The *orbits* of X are its orbits under the action of $\mathrm{Aut}(\mathcal{A})$. We say that X is *orbit-finite* if X has finitely many orbits.

The sets $\mathcal{A}$, $\mathcal{A}^n$ and $\mathcal{A}^{(n)}$, and the function s in Example 8 are equivariant. All sets in Example 9 are equivariant.

We will use the following results describing the action of $\mathrm{Aut}(\mathcal{A})$ on definable sets. All these results follow, using standard model-theoretic techniques (see e.g. [25]), from the evident fact the structure $\mathbb{A} = (\mathcal{A}, =)$ is *homogeneous*, i.e., every isomorphism between two finite substructures of $\mathbb{A}$ extends to an automorphism of $\mathbb{A}$. In particular, it admits *quantifier-elimination*, i.e., every first-order formula is equivalent to a quantifier-free formula.

Theorem 10. *Let X be a set definable by an expression without free variables. Then:*

1) *X is equivariant.*
2) *X is definable by a quantifier-free expression which can be computed in polynomial space from any expression defining X.*
3) *X is orbit-finite, and each of its orbits is definable by a quantifier-free expression.*
4) *If Y is an equivariant subset of X, then Y is definable by a quantifier-free expression.*

Proposition 11. *For definable sets X and Y, it is decidable in polynomial space whether $X \in Y$, $X \subseteq Y$, $X = Y$.*

A *system of orbit representatives* of X is a set R which contains exactly one element of each orbit of X.

Lemma 12. *One can compute, in polynomial space, a system of orbit representatives of a definable set X.*

C. Order-definable sets

We will sometimes find it advantageous to have a total order $<$ on $\mathcal{A}$, isomorphic to the ordering of the rational numbers. The development in Section II-A can be extended to this setting, by the use of the total order relation $<$ in first-order formulas that define sets with atoms. Sets and functions defined this way will be called *order-definable*. Clearly, any definable set is also order-definable. When we want to make clear that we do not allow the total order in the formulas, we can speak about *equality-definable* sets.

Example 13. The total order relation $< \subseteq \mathcal{A}^2$ is order-defined by the expression

$$\{(a, b) \mid a, b \in \mathcal{A}, a < b\}.$$

In contrast, no total ordering on $\mathcal{A}$ is equality-definable. □

Let $\mathrm{Aut}(\mathcal{A}, <)$ be the group of *monotone atom permutations*, i.e., automorphisms of the total order on $\mathcal{A}$. The notions of action, equivariance and orbits, can be developed as previously, with $\mathrm{Aut}(\mathcal{A})$ replaced by $\mathrm{Aut}(\mathcal{A}, <)$ throughout, and definable sets replaced by order definable sets. (This is a special case of a construction from [23], where various "atom symmetries" are considered, which is itself a special case of permutation models studied in set theory.) To distinguish from previous notions, we shall speak of *monotone-equivariant* sets and functions.

Note that the restriction to monotone permutations may cause the number of orbits in a set to grow. For example, the set $\mathcal{A}^{(n)}$, a single-orbit set under the action of $\mathrm{Aut}(\mathcal{A})$, decomposes into $n!$ orbits under the action of $\mathrm{Aut}(\mathcal{A}, <)$. However, an orbit-finite set remains orbit-finite under the action of monotone atom permutations.

All results from Section II-B hold for order-definable sets as well. For this, it is crucial that the structure $(\mathbb{Q}, <)$ is homogeneous. Indeed, for a different order such as that of natural or integer numbers, most of those results would fail.

III. Infinite instances

In this section we study the decidability and complexity of solving definable CSP instances with atoms, i.e., the decision problem CSP-Inf($\mathbb{T}$) described in the Introduction. Examples 1 and 2 only concern finite templates; we prove decidability for the more general case of *locally finite* templates, where every relation is finite.

In Section III-A the main result is Theorem 19, which states that the existence of a solution of a definable instance over a locally finite template can be decided. The key technical step towards it, Theorem 17, will also be of use in further sections. In Section III-B we focus on the case when the template $\mathbb{T}$ is finite, and prove matching lower and upper bounds on the complexity of the problem CSP-Inf($\mathbb{T}$).

Before we begin, observe that for an equivariant instance, the existence of a solution does not imply the existence of an equivariant solution.

Example 14. Consider an instance $\mathbb{I}$ with $\mathcal{A}^{(2)}$ as the set of variables, and with the set of constraints:

$$\big\{\big(((a, b), (b, a)), R\big) \mid a, b \in \mathcal{A}, a \neq b\big\},$$

where $R = \{(0, 1), (1, 0)\}$ is the inequality relation on the finite domain $\{0, 1\}$. It is easy to see that $\mathbb{I}$ has a solution. Indeed, for any distinct atoms a and b, one can arbitrarily assign a value 0 or 1 to the variable (a, b), and then assign the other value to (b, a). However, it is impossible to do so

in a way that would be invariant with respect to all atom permutations, therefore no equivariant solution exists.

On the other hand, there exists a monotone-equivariant solution, namely the assignment

$$f(a,b) = \begin{cases} 0 & \text{if } a < b \\ 1 & \text{otherwise.} \end{cases}$$

This anticipates Theorem 17. □

A. General decidability

For any instance $\mathbb{I}$ over a template $\mathbb{T}$, let $\hom(\mathbb{I},\mathbb{T})$ denote the set of solutions of $\mathbb{I}$. It is a subset of the set T^I of all functions from I to T. The latter set is equipped with the product topology, i.e., where basic open neighborhoods of a function $f : I \to T$ are of the form:

$$\mathcal{B}_J(f) = \{g : I \to T \mid g(x) = f(x) \text{ for all } x \in J\} \qquad (1)$$

for finite subsets $J \subseteq I$. This topology is also called the topology of *pointwise convergence* (where T is discrete), since a sequence of mappings $f_1, f_2, \ldots$ converges in this topology if and only if the sequence $f_1(v), f_2(v), \ldots$ stabilizes for every $v \in I$.

The following lemma extracts a crucial property of locally finite templates. We say that an instance is *constrained* if every variable in it appears in some constraint.

Lemma 15. *For any locally finite template $\mathbb{T}$, and any constrained instance $\mathbb{I}$ over $\mathbb{T}$, $\hom(\mathbb{I},\mathbb{T})$ is a compact subset of T^I.*

Similarly as above, the group $\mathrm{Aut}(\mathcal{A})$ of atom permutations inherits the structure of a topological space (in fact, a topological group, which is not even locally compact), as a subset of $\mathcal{A}^{\mathcal{A}}$ with the product topology.

Lemma 16. *For any definable, equivariant, constrained instance $\mathbb{I}$ over an equivariant template $\mathbb{T}$, $\mathrm{Aut}(\mathcal{A})$ acts continuously on $\hom(\mathbb{I},\mathbb{T})$.*

By definition, fixpoints of the action of $\mathrm{Aut}(\mathcal{A})$ on $\hom(\mathbb{I},\mathbb{T})$ are exactly equivariant solutions. Example 14 shows that the action may have no fixpoints. This changes if we assume $\mathcal{A}$ to be ordered and restrict to monotone atom permutations, as described in Section II-C. Indeed:

Theorem 17. *For any definable, equivariant, constrained instance $\mathbb{I}$ over an equivariant, locally finite template $\mathbb{T}$, if $\mathbb{I}$ has a solution than it has a monotone-equivariant solution.*

Proof. Note that $\mathrm{Aut}(\mathcal{A},<)$ is isomorphic to the automorphism group of the total order of rational numbers. It is also a subgroup of $\mathrm{Aut}(\mathcal{A})$, so by Lemma 16 it acts continuously on $\hom(\mathbb{I},\mathbb{T})$. By definition, the fixpoints of this action are exactly monotone-equivariant solutions. Now apply Pestov's theorem (Theorem 4) using Lemma 15. □

The last missing part for the decidability of CSP-Inf($\mathbb{T}$) is the following lemma, whose proof follows general principles of equivariant computation on orbit-finite structures for arbitrary atom symmetries, studied in [22], [26].

Lemma 18. *For any equivariant, locally finite template $\mathbb{T}$, it is decidable whether a given definable, constrained, equivariant instance $\mathbb{I}$ over $\mathbb{T}$ has a monotone-equivariant solution.*

Finally, we are ready to prove the main result of this section:

Theorem 19. *For any equivariant, locally finite template $\mathbb{T}$, it is decidable whether a given definable, equivariant $\mathbb{I}$ over $\mathbb{T}$ has a solution.*

Proof. Remove the variables of $\mathbb{I}$ which are not constrained and apply Theorem 17 and Lemma 18. □

Remark 20. The careful reader may notice that Example 2 from the Introduction does not quite fit the development presented so far. Indeed, the instance (i.e. the equation system) considered there is not equivariant, or definable by an expression without free variables: atoms $\underline{1}$ and $\underline{2}$ are singled out in it. However, it is definable by an expression with two free variables, say v_1, v_2, with a valuation val that maps them to $\underline{1}$ and $\underline{2}$ respectively. In terminology of [21], [23], the instance is *supported* by the set $\{\underline{1},\underline{2}\}$.

With a little effort, the results of this section can be generalized to all definable instances and templates, dropping the equivariance assumption. Indeed, let both $\mathbb{I}$ and $\mathbb{T}$ be definable by expressions with free variables and valuations, with other assumptions as before. Let $a_1, \ldots, a_n \in \mathcal{A}$ be the (finite) set of all atoms taken as values by either of the valuations. Lemma 15 holds with no change. Lemma 16 holds as well, with $\mathrm{Aut}(\mathcal{A})$ replaced by $\mathrm{Aut}(\mathcal{A}, a_1, \ldots, a_n)$, the group of those atom permutations that fix all the a_i.

Consider the group $\mathrm{Aut}(\mathcal{A}, <, a_1, \ldots, a_n)$ of those *monotone* atom permutations that fix all the a_i; by analogy to the equivariant case, this group acts continuously on solutions of $\mathbb{I}$, and its fixpoints are monotone solutions invariant with respect to all permutations in that group.

To re-prove Theorem 17, notice that $\mathrm{Aut}(\mathcal{A}, <, a_1, \ldots, a_n)$ is an open subgroup of $\mathrm{Aut}(\mathcal{A}, <)$ therefore, by [27, Lemma 13], Theorem 4 works for it as well. Finally, Lemma 18 and Theorem 19 are proved entirely analogously for all definable structures.

A more substantial generalization is also possible, where one replaces a mere set $\mathcal{A}$ by a *homogeneous* relational structure of atoms, along the lines of [23]. Supposing that $\mathcal{A}$ is a reduct of a so-called Ramsey structure with enough decidability properties, the above development can be repeated, with Pestov's theorem replaced by its generalization due to Kechris, Pestov and Todorcevic [7]. A more detailed description of this is deferred to a full version of this paper.

B. Finite templates

Consider now a classical, *finite* template $\mathbb{T}$, without atoms, such as in Examples 1 and 2. The algorithm for solving definable instances over $\mathbb{T}$ that arises from a general proof of Lemma 18 is double exponential. However, for finite templates this complexity can be lowered using the following PSPACE reduction to CSP($\mathbb{T}$):

Proposition 21. *For every equivariant, definable instance $\mathbb{I}$ over a finite template $\mathbb{T}$ one can compute (in polynomial space) a finite instance $\mathbb{I}^*$ of size exponential in the size of the set expression that defines $\mathbb{I}$, and such that $\mathbb{I}$ has a solution if and only if $\mathbb{I}^*$ has a solution.*

Proof. By Theorem 17, $\mathbb{I}$ has a solution if and only if it has a monotone-equivariant solution. We construct a finite instance $\mathbb{I}^*$ whose solutions correspond bijectively to monotone-equivariant solutions of $\mathbb{I}$. In the rest of the proof, by orbits we mean orbits with respect to monotone atom permutations.

Let e be the set expression that defines the instance $\mathbb{I}$. The set I^* of variables of $\mathbb{I}^*$ consists of the orbits of the set I of variables of $\mathbb{I}$. Their number is at most exponential in the size of e and they can be enumerated in polynomial space, by scanning all quantifier-free types of formulas with n free variables, where n is the number of variables in the expression e.

For every constraint $\big((x_1, \ldots, x_k), R\big)$ of $\mathbb{I}$ we take the orbits $O_1, \ldots, O_k$ of the variables $x_1, \ldots, x_k$, respectively, and add a constraint $\big((O_1, \ldots, O_k), R\big)$ to $\mathbb{I}^*$. The number of constraints is also at most exponential in the size of e.

Every monotone-equivariant function from I to T is constant on the orbits of I. Hence, it is easy to see that there is a bijective correspondence between solutions of $\mathbb{I}^*$ and monotone-equivariant solutions of $\mathbb{I}$. □

By analogy to Remark 20, the above result can be generalized to all definable instances, and hence gives an upper complexity bound of solving definable instances for each finite template $\mathbb{T}$: if CSP($\mathbb{T}$) is in a complexity class $\mathcal{C}$ such that $\mathrm{L} \subseteq \mathcal{C}$, then CSP-Inf($\mathbb{T}$) is in the "exponentially larger" class $\exp(\mathcal{C})$, defined by:

$$\exp(\mathcal{C}) = \bigcup_{k \in \mathbb{N}} \{L : \mathrm{pad}_k(L) \in \mathcal{C}\}$$

where, for a language L, the language $\mathrm{pad}_k(L)$ consists of words from L of any length m padded with arbitrary letters to the length 2^{m^k}. For example, $\exp(\mathrm{L}) = \textsc{PSpace}$ and $\exp(\mathrm{NP}) = \textsc{NExp}$.

As it turns out, this upper bound is tight:

Theorem 22. *For any $\mathcal{C}$ such that $\mathrm{L} \subseteq \mathcal{C}$, if* CSP($\mathbb{T}$) *is $\mathcal{C}$-hard then* CSP-Inf($\mathbb{T}$) *is $\exp(\mathcal{C})$-hard, under logarithmic space reductions.*

Together with Proposition 21, this proves Theorem 3. As examples, 3-colorability of finite graphs is NP-complete, so the same problem for definable graphs (see Example 1) is NExp-complete; solving finite systems of linear equations over $\mathbb{Z}_k$ is $\textsc{Mod}_k\mathrm{L}$-complete [28], so the same problem for definable systems of equations (see Example 2) is complete for the class $\exp(\textsc{Mod}_k\mathrm{L}) = \textsc{Mod}_k\textsc{PSpace}$ of those languages L for which there exists a nondeterministic polynomial space Turing machine M_L such that $w \in L$ if and only if the number of accepting runs of M_L on input w is divisible by k.

Remark 23. Our proof of Theorem 22 actually works already if CSP($\mathbb{T}$) is $\mathcal{C}$-complete under poly-logarithmic space reductions; completeness of $\mathcal{C}$ under logarithmic space reductions is not necessary. Note that it is an open problem whether the class of poly-log space algorithms is contained in PTime.

In a proof of Theorem 22 one must construct definable instances of CSP-Inf($\mathbb{T}$) from words in a language $L \in \exp(\mathcal{C})$. In our general proof, set expressions that define these instances use the full power of first order logic, including quantification over atoms. However, for all standard examples of templates $\mathbb{T}$, e.g. those that correspond to the 3-colorability problem, Boolean 3-satisfiability, solving linear equations over $\mathbb{Z}_k$ or Boolean Horn-satisfiability, we have found reductions that use the simplest possible formulas over atoms: comparing two atoms for equality. We are unable to find a general proof which would only use so simple set expressions.

IV. Finite instances

In this section we study the complexity of the problem CSP($\mathbb{T}$), where $\mathbb{T}$ is a fixed locally finite template. Recall that instances of this problem are finite structures. We demonstrate that many results known from the classical setting, where the templates are finite, lift to the locally finite setting. In Section IV-A we show that polymorphisms of $\mathbb{T}$ determine the complexity of CSP($\mathbb{T}$). In Section IV-B we present a general method of lifting results concerning finite templates to locally finite templates, and we show a few applications. Finally, in Section IV-C, we show that for a locally finite, definable template $\mathbb{T}$, the problem CSP($\mathbb{T}$) reduces (in polynomial time) to the problem CSP($\hat{\mathbb{T}}$) for a *finite* template $\hat{\mathbb{T}}$. Moreover, if $\mathbb{T}$ admits polymorphisms satisfying some linear identities, then so does $\hat{\mathbb{T}}$, which in many cases gives an upper bound on the complexity of CSP($\mathbb{T}$). To prove these results, we are sometimes forced to consider infinite templates $\mathbb{T}$ as CSP instances over other templates, and there the results of Section III come handy.

A. Algebraic foundations

We now generalize, to locally finite templates, several classical theorems concerning the relationship between the set of polymorphisms of a template, its set of pp-definable relations and its complexity, and constructions of core templates. Most of the proofs are standard, but they require a few tweaks and nontrivial observations.

1) Pp-formulas: A *primitive positive formula* (or *pp-formula*) is a first-order formula, possibly with free variables, which uses only existential quantification (no universal quantifiers), conjunctions (no disjunctions nor negations), and atomic formulas (no negations of atomic formulas), which might include equalities among variables. A typical pp-formula is

$$\phi(x, z, t) = \exists y. R(x, y) \wedge S(y, z) \wedge z = t.$$

If $\mathbb{T}$ is a relational structure and ϕ is a pp-formula over the signature of $\mathbb{T}$ with free variables $x_1, \ldots, x_n$, then we denote by $\phi(\mathbb{T}) \subseteq T^n$ the set of those tuples $(t_1, \ldots, t_n)$ which satisfy ϕ in $\mathbb{T}$. Any relation of the form $\phi(\mathbb{T})$, where ϕ is some pp-formula, is said to be a *pp-definable* relation of $\mathbb{T}$. By $\mathrm{ppDef}\,\mathbb{T}$ we denote the family of all *finite* pp-definable relations of $\mathbb{T}$.

A *generalized pp-formula* Φ is a pair $(\mathbb{I}, \alpha)$, where $\mathbb{I}$ is an instance (equivalently, a relational structure) and $\alpha : \{x_1, \dots, x_n\} \to I$ is a function from a finite set of *free variables* of Φ. We say that Φ is *finite* if the instance $\mathbb{I}$ has finitely many variables and finitely many constraints. If $\mathbb{T}$ is a relational structure over the same signature as $\mathbb{I}$, then Φ defines a relation on T of arity n:

$$\phi(\mathbb{T}) \stackrel{\text{def}}{=} \big\{\big(f(x_1), \dots, f(x_n)\big) \mid f \in \hom(\mathbb{I}, \mathbb{T})\big\}.$$

It is a standard result that finite generalized pp-formulas define the same relations as pp-formulas. The following lemma shows that for finite relations and locally finite templates, *arbitrary* generalized pp-formulas do not define anything more:

Lemma 24. *Let $\mathbb{T}$ be a locally finite template. The following conditions are equivalent for a finite relation $R \subseteq \mathbb{T}^n$*

1) *There is a pp-formula ϕ such that $\phi(\mathbb{T}) = R$,*
2) *There is a finite generalized pp-formula Φ such that $\Phi(\mathbb{T}) = R$.*
3) *There is a generalized pp-formula Φ such that $\Phi(\mathbb{T}) = R$.*

Lemma 24 relies on the following lemma, which is a variation of the compactness argument used in Lemma 15 (the union of structures is taken vertex-wise and edge-wise).

Lemma 25. *For any structure $\mathbb{I}$ and an ascending sequence $\mathbb{I}_0 \subseteq \mathbb{I}_1 \subseteq \mathbb{I}_2 \subseteq \dots \subseteq \mathbb{I}$ of substructures such that $\bigcup_i \mathbb{I}_i = \mathbb{I}$, and for any locally finite structure $\mathbb{T}$, there is a homomorphism $f : \mathbb{I} \to \mathbb{T}$ if and only if for each $n \geq 0$ there is a homomorphism $f_n : \mathbb{I}_n \to \mathbb{T}$. If all the f_n extend a single homomorphism $f_0 : \mathbb{I}_0 \to \mathbb{T}$, then so does f.*

Proposition 26. *Let $\mathbb{B}$ and $\mathbb{C}$ be locally finite, definable templates over the same domain B. If $\operatorname{ppDef}\mathbb{B} \subseteq \operatorname{ppDef}\mathbb{C}$ then* CSP($\mathbb{B}$) *reduces to* CSP($\mathbb{C}$)*, via a polynomial-time reduction.*

2) The Inv-Pol connection: An *operation* on T is a function $f : T^k \to T$, for some number k (the *arity*). We say that the operation f *preserves* a relation R of arity n on T if for any k tuples in R:

$$(x_{11}, \dots, x_{1n}), (x_{21}, \dots, x_{2n}), \dots, (x_{k1}, \dots, x_{kn}) \in R,$$

the tuple obtained from them by applying f componentwise:

$$\big(f(x_{11}, \dots, x_{k1}), f(x_{12}, \dots, x_{k2}), \dots, f(x_{1n}, \dots, x_{kn})\big)$$

belongs to R as well. We also say that R is *invariant* under the operation f.

If $\mathcal{F}$ is a family of operations of T, then by $\operatorname{Inv}\mathcal{F}$ we denote the set of relations which are invariant under every operation in $\mathcal{F}$. Dually, if $\mathcal{R}$ is a family of relations on T, then by $\operatorname{Pol}\mathcal{R}$ we denote the set of those operations that preserve all relations in $\mathcal{R}$. If $\mathbb{T}$ is a relational structure, then $\operatorname{Pol}\mathbb{T}$ is defined as $\operatorname{Pol}\mathcal{R}$, where $\mathcal{R}$ is the set of relations of $\mathbb{T}$, and elements of $\operatorname{Pol}\mathbb{T}$ are called *polymorphisms* of $\mathbb{T}$.

A polymorphism of $\mathbb{T}$ of arity n can be equivalently defined as follows. Equip $\mathbb{T}^n$ with the *cartesian product* structure, i.e., for a relation R of $\mathbb{T}$ of arity k and a k-tuple of n-tuples of $\mathbb{T}$,

$$R\big((x_{11}, \dots, x_{1n}), (x_{21}, \dots, x_{2n}), \dots, (x_{k1}, \dots, x_{kn})\big)$$

holds in $\mathbb{T}^n$ if and only if each $i = 1, \dots, n$, the relation $R(x_{1i}, x_{2i}, \dots, x_{ki})$ holds in $\mathbb{T}$. It is easy to see that a mapping $f : T^n \to T$ is a polymorphism if and only if it is a homomorphism from $\mathbb{T}^n$ to $\mathbb{T}$.

Proposition 27. *For any locally finite template $\mathbb{T}$, and any finite relation R over T, $R \in \operatorname{ppDef}\mathbb{T}$ if and only of $R \in \operatorname{Inv}\operatorname{Pol}\mathbb{T}$.*

Proposition 27 is analogous to a fundamental theorem of algebraic finite CSP theory [1]. In the proof of the "only if" part, we define an invariant relation using a generalized pp-formula, and apply Lemma 24.

Theorem 28. *For any two locally finite, definable templates $\mathbb{B}$ and $\mathbb{C}$ over the same domain B, if $\operatorname{Pol}\mathbb{C} \subseteq \operatorname{Pol}\mathbb{B}$ then* CSP($\mathbb{B}$) *reduces to* CSP($\mathbb{C}$) *via a polynomial-time reduction.*

Proof. If $\operatorname{Pol}\mathbb{C} \subseteq \operatorname{Pol}\mathbb{B}$ then $\operatorname{Inv}\operatorname{Pol}(\mathbb{C}) \supseteq \operatorname{Inv}\operatorname{Pol}(\mathbb{B})$, hence $\operatorname{ppDef}(\mathbb{C}) \supseteq \operatorname{ppDef}(\mathbb{B})$. Conclude using Proposition 26. □

Corollary 29. *If $\operatorname{Pol}\mathbb{B} = \operatorname{Pol}\mathbb{C}$ then* CSP($\mathbb{B}$) *and* CSP($\mathbb{C}$) *are equivalent, up to polynomial time reductions.*

3) Core structures: We say that a structure $\mathbb{T}$ is a *core* if every endomorphism of $\mathbb{T}$ is a monomorphism. If $\mathbb{T}$ is a structure and A is a finite subset of its domain, then by $\mathbb{T}|_A$ we denote the template with domain A, whose relations are the restrictions to A of all relations $R \in \operatorname{ppDef}\mathbb{T}$. In the proof of the implication (2→1), again we invoke Pestov's theorem (Theorem 4).

Proposition 30. *Let $\mathbb{T}$ be a monotone-equivariant, locally finite structure without isolated nodes. Then the following conditions are equivalent:*

1) *$\mathbb{T}$ is a core.*
2) *Every monotone-equivariant endomorphism of $\mathbb{T}$ is a monomorphism.*
3) *For any finite set $A \subseteq T$, the structure $\mathbb{T}|_A$ is a finite core.*

By analogy to the development in Section III, thanks to Proposition 30 template cores are computable:

Proposition 31. *Given a definable, equivariant, locally finite template $\mathbb{T}$, one can effectively test whether $\mathbb{T}$ is a core, and effectively construct a core which is a retract of $\mathbb{T}$.*

Proof (sketch). A *retraction* of $\mathbb{T}$ is an endomorphism r whose twofold composition $r; r$ is equal to r.

Test all monotone-equivariant retractions of $\mathbb{T}$. If there is such a retraction r which is not onto, then $\mathbb{T}$ is not a core. In this case, replace $\mathbb{T}$ by the image of r and repeat. □

B. From finite to locally finite templates

In this section, we present a general method of lifting results concerning finite templates to locally finite ones. This is achieved by observing that $\mathbb{T}$ can be covered by a family $\mathcal{F}$ of finite subtemplates (defined below), such that the question whether $\mathbb{T}$ admits polymorphisms satisfying a set of linear equations boils down to the question whether each finite template in $\mathcal{F}$ admits such polymorphisms.

Important classes of polymorphisms are defined using sets of identities that they satisfy; usually, those identities are of a specific shape. Formally, let Γ be a functional signature, i.e., a set of function names with associated finite arities. A *linear identity* is an expression of the form $r \approx s$, where r and s are either variables or Γ-terms with exactly one function symbol. A Γ-algebra *satisfies* an identity $r \approx s$ if for any valuation of the variables in r and s, both sides have the same value. For a set E of linear identities, we say that a template $\mathbb{T}$ *admits E-polymorphisms* if there is a Γ-algebra with universe T, whose operations are polymorphisms of $\mathbb{T}$ and which satisfies all identities in E.

For example, a *majority polymorphism* is a ternary polymorphism t that satisfies linear identities:

$$t(x,y,y) \approx t(y,x,y) \approx t(x,y,y) \approx y,$$

and a *Maltsev polymorphism* is a ternary one that satisfies linear identities:

$$t(x,y,y) \approx t(y,y,x) \approx x.$$

Other polymorphism classes defined by linear identities will be considered below.

Let $\mathbb{T} = (T, \mathcal{R})$ be a template and let $\mathcal{R}_0 \subseteq \mathcal{R}$ be some family of relations. The *subtemplate* of $\mathbb{T}$ induced by $\mathcal{R}_0$ is the template $\mathbb{T}_0 = (T_0, \mathcal{R}_0)$ whose domain T_0 consists of all those $x \in T$ which appear in some tuple that belongs to any of the relations in $\mathcal{R}_0$. We define a union $\mathbb{T}_1 \cup \mathbb{T}_2$ of two subtemplates $\mathbb{T}_1 = (T_1, \mathcal{R}_1)$ and $\mathbb{T}_2 = (T_2, \mathcal{R}_2)$ of $\mathbb{T}$ to be the template $(T_1 \cup T_2, \mathcal{R}_1 \cup \mathcal{R}_2)$, and analogously for unions of larger families of subtemplates.

Theorem 32. *Let $\mathbb{T}$ be a locally finite template and let $\mathbb{T}_1 \subseteq \mathbb{T}_2 \subseteq \ldots$ be an ascending sequence of subtemplates of $\mathbb{T}$ such that $\bigcup_i \mathbb{T}_i = \mathbb{T}$. If E is a set of linear identities then $\mathbb{T}$ admits E-polymorphisms if and only if for each i, the template $\mathbb{T}_i$ admits E-polymorphisms.*

Proof. The left-to-right implication is obvious, since any polymorphism of $\mathbb{T}$ is a polymorphism of all its subtemplates. To show the other implication we use a standard construction which allows to express a polymorphism of a template $\mathbb{T}$ as a solution of a CSP instance. Let E be a set of identities using function symbols from a functional signature Γ. We assume that there are no identities of the form $x \approx y$ where both x, y are either variables or constants, as the general case can be easily reduced to this one. For a template $\mathbb{T}$, define an instance $\mathbb{F}_E(\mathbb{T})$ as a disjoint union of instances as follows:

$$F_E(\mathbb{T}) = \coprod_{g \in \Gamma} \mathbb{T}_g,$$

where $\mathbb{T}_g$ is the n-fold cartesian product of $\mathbb{T}$, for n the arity of g. Furthermore, extend $F_E(\mathbb{T})$ by constraints as described below. For each identity in E, consider two possibilities:

- If the identity is of the form $g(\bar{x}) \approx g'(\bar{y})$, with $g, g' \in \Gamma$, then for every valuation $\mathrm{val} : V \to \mathbb{T}$ of the variables V in the tuples $\bar{x}$ and $\bar{y}$, add a binary constraint $\big((\bar{x}[\mathrm{val}], \bar{y}[\mathrm{val}]), =\big)$, where $\bar{x}[\mathrm{val}] \in \mathbb{T}_g$ and $\bar{y}[\mathrm{val}] \in \mathbb{T}_{g'}$.
- If the identity is of the form $g(\bar{x}) \approx u$ or $u \approx g(\bar{x})$, with $g \in \Gamma$ and u a variable or a constant, then for each valuation $\mathrm{val} : V \to \mathbb{T}$ of the variables V in $\bar{x}$ and u, add a unary constraint $\big(\bar{x}[\mathrm{val}], \{u[\mathrm{val}]\}\big)$.

Let $\overline{\mathbb{T}}$ be the structure $\mathbb{T}$ equipped additionally with the singleton unary relations and the relation $=$. The instance $F_E(\mathbb{T})$ is over $\overline{\mathbb{T}}$, and it has a solution if and only if $\mathbb{T}$ admits E-polymorphisms.

Suppose now that $\mathbb{T}$ is locally finite and let $\mathbb{T}_1 \subseteq \mathbb{T}_2 \subseteq \ldots$ be an ascending sequence of subtemplates of $\mathbb{T}$ such that $\bigcup_i \mathbb{T}_i = \mathbb{T}$, and for each i the template $\mathbb{T}_i$ admits E-polymorphisms. This means that for every i there exists a solution of the instance $F_E(\mathbb{T}_i)$ (which can be seen as an instance over $\overline{\mathbb{T}}$). The corresponding sequence of instances $(F_E(\mathbb{T}_i))_i$ is ascending and satisfies $\bigcup F_E(\mathbb{T}_i) = F_E(\mathbb{T})$. It follows from Lemma 25 that there exists a solution of the instance $F_E(\mathbb{T})$. Hence $\mathbb{T}$ admits E-polymorphisms. □

Remark 33. Note that if E is a finite set of linear identities and the template $\mathbb{T}$ is definable, then so is $F_E(\mathbb{T})$. By Theorem 19, it can be decided whether $\mathbb{T}$ admits E-polymorphisms.

In the literature, there are two general algorithmic principles for solving CSP($\mathbb{T}$) in polynomial time for wide classes of finite templates $\mathbb{T}$. Theorem 32 lets us translate algebraic characterizations of those classes to the setting of locally finite templates.

The first algorithm can be seen as a generalization of Gaussian elimination, and it is based on the fact that for every finite instance $\mathbb{I}$ over $\mathbb{T}$ the set of solutions has a "small" representing set. For a finite structure $\mathbb{A}$ let $s_{\mathbb{A}}(n)$ denote the logarithm, base 2, of the number of all different pp-definable relations of arity n in $\mathbb{A}$. We say that a template $\mathbb{T}$ has *few subpowers* if there is a natural number k such that for every finite subtemplate $\mathbb{B}$ of $\mathbb{T}$ we have that $s_{\mathbb{B}}(n)$ is bounded from above by a polynomial of degree k. It is known [29], [30] that a finite template $\mathbb{T}$ has few subpowers if and only if $\mathbb{T}$ admits a k-edge polymorphism, where a *k-edge* polymorphism e is a $k+1$-ary polymorphism that satisfies the identities

$$e(x,x,y,y,y,\ldots,y,y) \approx e(x,y,x,y,y,\ldots,y,y) \approx y$$

and

$$e(y,y,y,x,y,\ldots,y,y) \approx e(y,y,y,y,x,\ldots,y,y) \approx \ldots$$
$$\ldots \approx e(y,y,y,y,y,\ldots,x,y) \approx e(y,y,y,y,y,\ldots,y,x) \approx y.$$

Corollary 34. *A locally finite template $\mathbb{T}$ has few subpowers if and only if $\mathbb{T}$ admits a k-edge polymorphism for some $k > 0$.*

Moreover, k-edge polymorphisms characterize in some sense (see [29]) all locally finite templates that can by solved in polynomial time by a "Gaussian-like" algorithm.

The second algorithm determines whether a solution of a given instance exists by looking at subinstances of bounded size and checking if there is a consistent set of local solutions. A template of *bounded-width* is a template for which such an algorithm, called the *local consistency algorithm*, is correct. It follows from the results of [31]–[33] that a core finite template $\mathbb{T}$ has bounded width if and only if $\mathbb{T}$ admits weak near unanimity polymorphisms v of arity 3 and w of arity 4 that satisfy $v(y,x,x) \approx w(y,x,x,x)$, where a *weak near unanimity* polymorphism t is one that satisfies the identities

$$t(x,x,\dots,x) \approx x \text{ and}$$
$$t(y,x,\dots,x) \approx t(x,y,x,\dots,x) \approx \dots \approx t(x,x,\dots,x,y).$$

Corollary 35. *A core, locally finite template $\mathbb{T}$ has bounded width if and only if $\mathbb{T}$ admits weak near unanimity polymorphisms v of arity 3 and w of arity 4 that satisfy $v(y,x,x) \approx w(y,x,x,x)$. Moreover, it can be decided whether a definable, locally finite template $\mathbb{T}$ has bounded width.*

Proof (sketch). We show the harder "only if" part. Suppose that $\mathbb{T}$ has bounded width. Consider a finite set $A \subseteq T$. By Proposition 30, $\mathbb{T}|_A$ is a finite core and has bounded width. By the results of finite CSP theory, $\mathbb{T}|_A$ admits polymorphisms v_A, w_A satisfying the required linear identities. Since A is arbitrary, we may apply Theorem 32, obtaining polymorphisms v, w of $\mathbb{T}$, which also satisfy the required identities.

The last part of the statement of the corollary follows from Proposition 31 and Remark 33. □

Purely algebraic results, which show for instance that the existence of polymorphisms of some kind ensures a finite template $\mathbb{T}$ to admit some other polymorphisms can also be translated to the locally finite setting using Proposition 32. The following is an easy consequence of Lemma 9 of [34]:

Corollary 36. *If a core, locally finite template $\mathbb{T}$ has a Maltsev polymorphism and has bounded width, then it has a majority polymorphism.*

Many other results can be proved following the same pattern.

C. From locally finite to finite templates

This section is a proof of the following theorem:

Theorem 37. *For every equivariant, definable, locally finite template $\mathbb{T}$ there is a finite template $\hat{\mathbb{T}}$ such that:*

- *for every finite instance $\mathbb{I}$ over $\mathbb{T}$ there is a finite instance $\hat{\mathbb{I}}$ over $\hat{\mathbb{T}}$:*
  - *computable from $\mathbb{I}$ in polynomial time, and*
  - *such that $\mathbb{I}$ has a solution if and only if $\hat{\mathbb{I}}$ has a solution,*
- *for every set of linear identities E, if $\mathbb{T}$ admits E-polymorphisms then $\hat{\mathbb{T}}$ admits E-polymorphisms.*

We assume atoms to be totally ordered as described in Section II-C. By $\underline{i}$ we denote the atom that corresponds to the rational number i. All orbits below are considered with respect to monotone atom permutations.

Fix any order-definable set $x = e[\mathrm{val}]$, where e is an expression and $\mathrm{val} : V \to \mathcal{A}$ a valuation. There is a unique order-preserving bijection f between the range of val and the set $\{\underline{1}, \dots, \underline{n}\}$, where n is the number of elements in the range of val. By $[x]$ we denote the set defined by $e[\mathrm{val}; f]$. Note that $[x] = [y]$ if and only if they are in the same orbit. The set $[x]$ can therefore be seen as a representative of this orbit.

Assume without loss of generality that all instances are constrained. For any instance $\mathbb{I}$ over $\mathbb{T}$ and a variable $x \in I$, let $\mathbb{I}_x$ be the largest constrained subinstance of $\mathbb{I}$ such that x belongs to the tuple of variables in every constraint of $\mathbb{I}_x$. By $U_{(\mathbb{I},x)}$ we denote the unary predicate over $\mathbb{T}$ defined by the generalized pp-formula $(\mathbb{I}_x, \alpha_x)$, where $\alpha_x : \{x\} \to \mathbb{I}_x$ is the inclusion mapping (we write simply U_x whenever the instance $\mathbb{I}$ is clear from the context). Observe that if $f : \mathbb{I} \to \mathbb{T}$ is a solution of the instance $\mathbb{I}$ then $f(x) \in U_{(\mathbb{I},x)}$. We say that an assignment $f : I \to T$ is *feasible* if it maps every x to an element of $U_{(\mathbb{I},x)}$. To decide whether $\mathbb{I}$ has a solution, it is enough to consider feasible assignments.

Fix a constrained instance $\mathbb{I}$ over $\mathbb{T}$. For each variable $x \in I$, let g_x be a monotone atom permutation that maps U_x to $[U_x]$. We define an instance $\hat{\mathbb{I}}$ as follows:

- the variables of $\hat{\mathbb{I}}$ are the variables of $\mathbb{I}$,
- for every constraint $\big((x_1,\dots,x_n), R\big)$ in $\mathbb{I}$ there is a constraint $\big((x_1,\dots,x_n), \hat{R}\big)$ in $\hat{\mathbb{I}}$, where

$$\hat{R} = g\big(R \cap (U_{x_1} \times \dots \times U_{x_n})\big),$$
$$g(t_1,\dots t_n) = (g_{x_1}(t_1), \dots, g_{x_n}(t_n)).$$

  Note that $\hat{R} \subseteq [U_{x_1}] \times \dots \times [U_{x_n}]$.

It is immediate from the above construction that for any instance $\mathbb{I}$, there is a bijective correspondence between feasible assignments for $\mathbb{I}$ and assignments for $\hat{\mathbb{I}}$: if a feasible assignment f for $\mathbb{I}$ maps a variable x to some $t \in U_x$, then the corresponding assignment for $\hat{\mathbb{I}}$ maps x to $g_x(t) \in [U_x]$. Moreover, a feasible assignment for $\mathbb{I}$ is a solution if and only if the corresponding assignment for $\hat{\mathbb{I}}$ is a solution. It follows that $\mathbb{I}$ has a solution if and only if $\hat{\mathbb{I}}$ has a solution.

We now define a finite template $\hat{\mathbb{T}}$ such that for any instance $\mathbb{I}$ over $\mathbb{T}$, the instance $\hat{\mathbb{I}}$ is over $\hat{\mathbb{T}}$. The domain of $\hat{\mathbb{T}}$ is the union:

$$\hat{T} = \bigcup_{(\mathbb{I},x)} [U_{(\mathbb{I},x)}],$$

where $(\mathbb{I}, x)$ ranges over all constrained instances $\mathbb{I}$ over $\mathbb{T}$, and their variables x. The relations of $\hat{\mathbb{T}}$ are all the relations which appear in the constraints of all instances of the form $\hat{\mathbb{I}}$. Notice that if the $\mathbb{T}$ is definable then the template $\hat{\mathbb{T}}$ is finite. Indeed, since $\mathbb{T}$ has finitely many orbits, the number of elements in the unary predicates $U_{(\mathbb{I},x)}$ is bounded, and hence they form an orbit-finite set. The domain $\hat{T}$ is the sum of their representatives, so it is finite. Finally, the arity of relations in $\hat{\mathbb{T}}$ is bounded by the maximal arity of a relation in $\mathbb{T}$, so there are finitely many possible relations.

This proves half of Theorem 37; we now prove the second half.

Lemma 38. *Let E be a set of linear identities. An equivariant, definable, locally finite template $\mathbb{T}$ admits E-polymorphisms if and only if it admits monotone-equivariant E-polymorphisms.*

Proof. In the proof of Theorem 32, an instance $\mathbb{F}_E(\mathbb{T})$ is constructed whose solutions correspond to E-polymorphisms of $\mathbb{T}$. Let $\mathbb{I}$ be its largest constrained subinstance. By Theorem 17, if $\mathbb{I}$ has a solution then it has a monotone-equivariant solution f. Since the domain of $\mathbb{F}_E(\mathbb{T})$ is a disjoint union of sets of the form T^n, it is possible to extend this solution to a monotone-equivariant solution of $\mathbb{F}_E(\mathbb{T})$, for example by defining $f(x)$ to be the projection to the first coordinate whenever $x \notin \mathbb{I}$. This solution corresponds to a monotone-equivariant E-polymorphism of $\mathbb{T}$. □

We now finish the proof of Theorem 37. For any set of linear identities E, let A be an algebra with universe T, whose operations are polymorphisms of $\mathbb{T}$, and such that A satisfies the identities in E. By Lemma 38 we can assume that the operations of A are monotone-equivariant. We define an algebra $\hat{A}$ with universe $\hat{T}$, whose operations are polymorphisms of $\hat{\mathbb{T}}$, and such that $\hat{A}$ satisfies the identities in E. Observe that the universe $\hat{T}$ is a subset of the universe T. Let $r : T \to \hat{T}$ be a function which is the identity on $\hat{T}$, and maps all the other elements of T to some fixed element t of $\hat{T}$. For every operation f of A, define the corresponding operation $\hat{f}$ of $\hat{A}$ by $\hat{f} = f; r$. Since all identities in E are linear, it is easy to see that $\hat{A}$ satisfies them.

It remains to show that every $\hat{f}$ is a polymorphism of $\hat{\mathbb{T}}$. To this end, pick an n-ary relation $\hat{R}$ of $\hat{\mathbb{T}}$. It follows from the construction of the template $\hat{\mathbb{T}}$ that there exists a pp-definable relation R in $\mathbb{T}$ such that $\hat{R} = F(R)$, where $F : \mathcal{A}^n \to \mathcal{A}^n$ is a tuple of monotone atom permutations. The relation R is invariant under the monotone-equivariant f, hence it is easy to see that so is $\hat{R}$. Moreover, since R is a subset of $\hat{T}^n$, it follows that it is invariant under $\hat{f}$. □

V. Order-invariant logics

In a previous paper [10], we studied the expressive power of various logics over a certain class of structures, which we now recall in a slightly simplified version. As an application of the methods developed in this paper, we briefly say how the main result of [10] can be re-developed in the setting of locally finite CSPs, showing the key ideas of the proof much more clearly and clearing them from a clutter of technicalities.

For a fixed, finite graph $\mathfrak{p}$, a *linearly $\mathfrak{p}$-patched structure* is a finite graph $\mathbb{G}$, together with a linearly ordered family $\mathfrak{p}_1 < \ldots < \mathfrak{p}_n$ of subgraphs of $\mathbb{G}$ (called *patches*), each of which is isomorphic to $\mathfrak{p}$, and which cover $\mathbb{G}$, i.e.,

$$\mathbb{G} = \mathfrak{p}_1 \cup \ldots \cup \mathfrak{p}_n.$$

For example, if $\mathfrak{p}$ is the 2-clique, then a linearly $\mathfrak{p}$-patched structure is the same as a graph together with a linear ordering of its edges.

First order formulas can be evaluated on linearly $\mathfrak{p}$-patched structures, allowing quantification $\forall v, \exists v$ over the vertices, $\forall \mathfrak{q}, \exists \mathfrak{q}$ over the patches, comparison $\mathfrak{q} < \mathfrak{q}'$ of the patches with respect to their linear ordering, and tests $\mathfrak{p} \models E(v, w)$ for each patch $\mathfrak{p}$ and vertices v, w (where E denotes the edge predicate). Various extensions of first order logic can be also evaluated over linearly $\mathfrak{p}$-patched structures, in particular the *Least Fix Point* logic (*LFP*), a well studied extension of first order logic by a fixpoint operation [8], [35]. It turns out that over linearly $\mathfrak{p}$-patched structures, LFP is equivalent to LFP+C (a further extension by a counting mechanism) and to polynomial time *Turing Machines with Atoms* – an analogue of Turing machines in the realms of sets with atoms [10], [24].

On the other hand, classical Turing machines can be evaluated on linearly $\mathfrak{p}$-patched structures, using standard bit-string encodings of relational structures. We say that LFP is equally expressive as PTIME over a class of structures $\mathcal{C}$, if every property of structures in $\mathcal{C}$ which is decidable in polynomial time, can be expressed by a formula of LFP.

The famous Immerman-Vardi [36], [37] and Cai-Fürer-Immerman [8] theorems can be reformulated as follows.

Theorem 39. *If $\mathfrak{p}$ is the 2-clique, then LFP is equally expressive as* PTIME *over linearly $\mathfrak{p}$-patched structures.*

Theorem 40. *There is a graph $\mathfrak{p}$, namely the disjoint union of two 3-cliques, such that LFP+C is less expressive than* PTIME *over linearly $\mathfrak{p}$-patched structures.*

A corollary of the main result of [10] is the following.

Theorem 41. *Given a graph $\mathfrak{p}$, it can be effectively decided whether LFP is equally expressive as* PTIME *over linearly $\mathfrak{p}$-patched structures.*

The proof of the theorem proceeds in several steps, and starts with the following observation.

Lemma 42. *LFP is equally expressive as* PTIME *over linearly $\mathfrak{p}$-patched structures if and only if it can test their isomorphism.*

Consider a pair of linearly $\mathfrak{p}$-patched structures $\mathbb{G}, \mathbb{G}'$, whose vertices are atoms, both with n patches denoted $\mathfrak{p}_1 < \ldots < \mathfrak{p}_n$ and $\mathfrak{q}_1 < \ldots < \mathfrak{q}_n$, respectively. An isomorphism from $\mathbb{G}$ to $\mathbb{G}'$ must map each $\mathfrak{p}_i$ isomorphically to $\mathfrak{q}_i$, for $i = 1, \ldots, n$. Define a finite instance $\mathbb{I}$ with variables $1, 2, \ldots, n$, and for each $1 \leq i, j \leq n$, a binary constraint

$$((i, j), C^{\mathfrak{p},\mathfrak{p}'}_{\mathfrak{r},\mathfrak{r}'}),$$

such that $C^{\mathfrak{p},\mathfrak{p}'}_{\mathfrak{r},\mathfrak{r}'}$ is the set of pairs (α, β) where $\alpha : \mathfrak{p}_i \to \mathfrak{q}_i, \beta : \mathfrak{p}_j \to \mathfrak{q}_j$ are isomorphisms which are consistent, i.e., $\alpha(v) = \beta(v)$ for all v such that both $v \in \mathfrak{p}_i$ and $v \in \mathfrak{p}_j$.

Clearly, there is an isomorphism of linearly $\mathfrak{p}$-patched structures from $\mathbb{G}$ to $\mathbb{G}'$ iff the instance $\mathbb{I}$ has a solution.

It is useful to consider a single template $\mathbb{T}_\mathfrak{p}$ such that any instance $\mathbb{I}$ obtained from two linearly $\mathfrak{p}$-patched structures $\mathbb{G}, \mathbb{G}'$ as above, is over $\mathbb{T}_\mathfrak{p}$. The domain of $\mathbb{T}_\mathfrak{p}$ consists of isomorphisms $\alpha : \mathfrak{q} \to \mathfrak{q}'$ between graphs isomorphic to $\mathfrak{p}$, whose vertices are atoms. For each quadruple $\mathfrak{q}, \mathfrak{q}', \mathfrak{r}, \mathfrak{r}'$, there is the binary relation $C^{\mathfrak{p},\mathfrak{p}'}_{\mathfrak{r},\mathfrak{r}'}$ defined above. $\mathbb{T}_\mathfrak{p}$ is equivariant

and, since every such relation is finite, it is locally finite. It is also not difficult to see that this structure is definable.

The next lemma follows from the results in [10].

Lemma 43. *LFP can test isomorphism of linearly $\mathfrak{p}$-patched structures if and only if the template $\mathbb{T}_{\mathfrak{p}}$ has bounded width.*

In [10], the proof of Theorem 41 then proceeds by constructing a finite template which corresponds to $\mathbb{T}_{\mathfrak{p}}$, in an ad-hoc way roughly similar to the one described in Section IV-C, and then further studying its properties. Since the construction of this finite template is technical, its study becomes obfuscated. Instead, using the results from the present paper, we can now work directly with the template $\mathbb{T}_{\mathfrak{p}}$, as sketched below.

By Corollary 35, it can be effectively tested whether $\mathbb{T}_{\mathfrak{p}}$ has bounded width. However, a more effective test is possible, due to the straightforward observation that the template $\mathbb{T}_{\mathfrak{p}}$ has a Maltsev polymorphism, defined by

$$M(\alpha,\beta,\gamma) = \begin{cases} \alpha \cdot \beta^{-1} \cdot \gamma & \text{if } \alpha,\beta,\gamma : \mathfrak{q} \to \mathfrak{q}' \text{ for some} \\ & \quad \mathfrak{q},\mathfrak{q}' \text{ isomorphic to } \mathfrak{p}, \\ \alpha & \text{otherwise.} \end{cases}$$

The following characterization then follows from Corollary 36.

Lemma 44. *The template $\mathbb{T}_{\mathfrak{p}}$ has bounded width if and only if it has a majority polymorphism.*

By Theorem 19, this last condition can be effectively tested. Lemmas 42, 43, 44 prove Theorem 41.

We believe that definable locally finite templates might arise naturally in other applications to Descriptive Complexity: although any such template is computationally equivalent to a finite one by Theorem 37, the presented reduction heavily relies on the ordering of the universe, and therefore cannot be mimicked by order-invariant logics, such as LFP.

ACKNOWLEDGMENTS

We are grateful to Manuel Bodirsky, Jakub Bulin and Marcin Kozik for their patience in answering our questions about various aspects of CSP theory. We also thank the anonymous reviewers for their thorough and helpful comments.

REFERENCES

[1] P. Jeavons, D. Cohen, and M. Gyssens, "Closure properties of constraints," *J. ACM*, vol. 44, no. 4, pp. 527–548, Jul. 1997.

[2] A. Bulatov and V. Dalmau, "A simple algorithm for mal'tsev constraints," *SIAM Journal on Computing*, vol. 36, no. 1, pp. 16–27, 2006. [Online]. Available: http://dx.doi.org/10.1137/050628957

[3] A. Bulatov, P. Jeavons, and A. Krokhin, "Classifying the complexity of constraints using finite algebras," *SIAM Journal on Computing*, no. 34, pp. 720–742, 2005.

[4] A. Bulatov, A. Krokhin, and P. Jeavons, "Constraint satisfaction problems and finite algebras," in *Proceedings of ICALP'00*, no. 1853, 2000, pp. 272–282, longer version available as an OUCL Technical Report : http://web.comlab.ox.ac.uk/oucl/publications/tr/tr-4-99.html.

[5] L. Barto and M. Kozik, "Absorbing subalgebras, cyclic terms, and the constraint satisfaction problem," *Log. Meth. Comp. Sci.*, vol. 8(1), 2012.

[6] V. Pestov, "On free actions, minimal flows, and a problem by Ellis," *Trans. Amer. Math. Soc.*, vol. 350, pp. 4149–4165, 1998.

[7] A. Kechris, V. Pestov, and S. Todorcevic, "Fraïssé limits, Ramsey theory, and topological dynamics of automorphism groups," *Geometric & Functional Analysis GAFA*, vol. 15, no. 1, pp. 106–189, 2005.

[8] J. Cai, M. Fürer, and N. Immerman, "An optimal lower bound on the number of variables for graph identifications," *Combinatorica*, vol. 12, no. 4, pp. 389–410, 1992.

[9] L. Barto and M. Pinsker, "The basic CSP reductions revisited," announced Nov. 2014. [Online]. Available: http://www.karlin.mff.cuni.cz/~barto/Articles/Banff_Barto.pdf

[10] B. Klin, S. Lasota, J. Ochremiak, and S. Toruńczyk, "Turing machines with atoms, constraint satisfaction problems, and descriptive complexity," in *Procs. of CSL-LICS'14*, 2014, pp. 58:1–58:10.

[11] M. H. Freedman, "K-sat on groups and undecidability," in *Procs. STOC*, ser. STOC '98, 1998, pp. 572–576.

[12] H. Chen, "Periodic constraint satisfaction problems: Tractable subclasses," *Constraints*, vol. 10, no. 2, pp. 97–113, 2005.

[13] S. S. Dantchev and F. D. Valencia, "On the computational limits of infinite satisfaction," in *Procs. (SAC)*, 2005, pp. 393–397.

[14] M. Bodirsky and J. Nesetril, "Constraint satisfaction with countable homogeneous templates," *J. Log. Comput.*, vol. 16, no. 3, pp. 359–373, 2006.

[15] M. Bodirsky, "Cores of countably categorical structures," *Logical Methods in Computer Science*, vol. 3, no. 1, 2007.

[16] M. Bodirsky and M. Pinsker, "Topological birkhoff," *CoRR*, vol. abs/1203.1876, 2012. [Online]. Available: http://arxiv.org/abs/1203.1876

[17] M. Bodirsky and J. Kára, "The complexity of temporal constraint satisfaction problems," *J. ACM*, vol. 57, no. 2, 2010.

[18] M. Bodirsky and M. Pinsker, "Schaefer's theorem for graphs," in *Procs. STOC*, 2011, pp. 655–664.

[19] M. Bodirsky and V. Dalmau, "Datalog and constraint satisfaction with infinite templates," *J. Comput. Syst. Sci.*, vol. 79, no. 1, pp. 79–100, 2013.

[20] M. Bodirsky and M. Pinsker, "Reducts of ramsey structures," *CoRR*, vol. abs/1105.6073, 2011.

[21] A. M. Pitts, *Nominal Sets: Names and Symmetry in Computer Science*. Cambridge University Press, 2013, vol. 57.

[22] M. Bojańczyk and S. Toruńczyk, "Imperative programming in sets with atoms," in *Procs. FSTTCS 2012*, ser. LIPIcs, vol. 18, 2012, pp. 4–15.

[23] M. Bojańczyk, B. Klin, and S. Lasota, "Automata theory in nominal sets," *Log. Meth. Comp. Sci.*, vol. 10, 2014.

[24] M. Bojanczyk, B. Klin, S. Lasota, and S. Torunczyk, "Turing machines with atoms," in *LICS*, 2013, pp. 183–192.

[25] W. Hodges, *Model Theory*, ser. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1993, no. 42.

[26] M. Bojańczyk, L. Braud, B. Klin, and S. Lasota, "Towards nominal computation," in *Procs. POPL 2012*, 2012, pp. 401–412.

[27] M. Bodirsky, M. Pinsker, and T. Tsankov, "Decidability of definability," *The Journal of Symbolic Logic*, vol. 78, no. 04, pp. 1036–1054, 2013.

[28] G. Buntrock, C. Damm, U. Hertrampf, and C. Meinel, "Structure and importance of logspace-mod-classes." in *Procs. STACS*, ser. Lecture Notes in Computer Science, vol. 480, 1991, pp. 360–371.

[29] P. Idziak, P. Markovic, R. McKenzie, M. Valeriote, and R. Willard, "Tractability and learnability arising from algebras with few subpowers," in *Logic in Computer Science, 2007. LICS 2007. 22nd Annual IEEE Symposium on*, July 2007, pp. 213–224.

[30] J. Berman, P. Idziak, P. Markovic, R. McKenzie, M. Valeriote, and R. Willard, "Varieties with few subalgebras of powers," *Transactions of The American Mathematical Society*, vol. 362, pp. 1445–1473, 2009.

[31] B. Larose and L. Zádori, "Bounded width problems and algebras," *Algebra universalis*, vol. 56, no. 3-4, pp. 439–466, 2007.

[32] L. Barto and M. Kozik, "Constraint satisfaction problems solvable by local consistency methods," *J. ACM*, vol. 61, no. 1, pp. 3:1–3:19, Jan. 2014.

[33] M. Kozik, A. Krokhin, M. Valeriote, and R. Willard, "Characterizations of several Maltsev conditions," *Algebra Universalis*, 2015, to appear.

[34] V. Dalmau and B. Larose, "Maltsev + datalog –> symmetric datalog," in *Proceedings of the Twenty-Third Annual IEEE Symposium on Logic in Computer Science (LICS 2008)*. IEEE Computer Society Press, June 2008, pp. 297–306.

[35] N. Immerman, *Descriptive complexity*, ser. Graduate texts in computer science. Springer, 1999.

[36] ——, "Upper and lower bounds for first order expressibility," *J. Comput. Syst. Sci.*, vol. 25, no. 1, pp. 76–98, 1982.

[37] M. Y. Vardi, "The complexity of relational query languages (extended abstract)," in *Procs. STOC*, 1982, pp. 137–146.