Karp's NP-Complete Problems over First-Order Definable Structures

Aidan Healy[0009−0003−1015−9437] and Bartek Klin[0000−0001−5793−7425]

University of Oxford, UK
{aidan.healy,bartek.klin}@cs.ox.ac.uk

Abstract. We determine the decidability of Karp's NP-complete problems on structures which are first-order definable over the theory of equality, also known as orbit-finite sets with atoms or nominal sets.

Keywords: Sets with atoms · Nominal sets · NP-complete problems

1 Introduction

We wish to take a range of classical decision problems that are usually considered for finite structures, and study them on the class of structures which are infinite but definable by first-order formulas that use equality only. Precise definitions will follow, but let us begin with two illustrative examples. Here and in the following, fix a countably infinite set $\mathbb{A}$, whose elements we call atoms.

Example 1. Consider $X = \binom{\mathbb{A}}{2}$, the set of two-element sets of atoms, and the family $\mathcal{S}$ of all three-element subsets of X of the form:

$$\{\{a, b\}, \{a, c\}, \{b, c\}\} \qquad \text{for all distinct } a, b, c \in \mathbb{A}.$$

This family admits an *exact cover*: there is a sub-family of $\mathcal{S}$ that forms a partition of X. However, to find such a cover we must break the pleasant symmetry of $\mathcal{S}$. For example, we may fix an enumeration of X, and proceed by induction starting with the empty family: in each step, take the first $\{a, b\}$ that has not been covered yet, choose some c such that neither $\{a, c\}$ nor $\{b, c\}$ have been covered, and add this to the family. The limit of this process is an exact cover.

Example 2. Now let X be the disjoint union of $\binom{\mathbb{A}}{2}$ and $\mathbb{A}$. The family $\mathcal{S}$ of all subsets of X of the form:

$$\{\{a, b\}, \{a, c\}, a\} \qquad \text{for all distinct } a, b, c \in \mathbb{A},$$

does *not* admit an exact cover of X. To see why, notice that to cover a pair $\{a, b\}$ we must include a set that contains either a or b. Any atom in $\mathbb{A}$ can be included only once in this way, so it can be used to cover only two pairs. More generally, k atoms can be used to cover only $2k$ pairs. For $k = 6$, this is not enough to cover all the $\binom{6}{2} = 15$ pairs that are built of the k atoms and need to be covered.

Both these arguments are quite simple, but substantial enough to make one wonder whether or not the classical problem ExactCover is decidable for structures of this kind. Indeed, one of our main results is that it is not.

By "structures of this kind" we mean structures (graphs, hypergraphs, formulas, families...) where all components and relations are defined by first-order formulas that only compare atoms for equality. Such structures are usually infinite, but they are finite up to bijective renaming of atoms. We will find it convenient to describe them within the framework of *sets with atoms*, but in the parlance of model theory (see e.g. [15]), they are simply relational structures which are first-order interpretable in $(\mathbb{A}, =)$.

The purpose of this paper is to determine the decidability, over such structures, of appropriately extended versions of the classical NP-complete problems listed by Karp in 1972 [16].

The idea of transporting computational problems from finite to infinite structures has been explored in several variants, with the main difference being the class of infinite structures considered. The broadest setting is that of *recursive structures* [13,14], where nodes are natural numbers and arbitrary decidable relations are permitted. Of course, in this setting all nontrivial questions become undecidable, and the focus is on determining where particular problems lie in the arithmetical hierarchy.

A smaller class is that of *automatic structures* [18,2,12], where nodes are represented as words over a finite alphabet, and relations are recognisable by multi-tape finite-state automata that read those words in parallel. For this class, the model checking problem for first-order logic extended with a limited form of second-order quantification, is decidable [24]. As a result, natural extensions of a few of Karp's problems become decidable [23,24], including Clique, SetPacking and a variant of SetCover. Several other problems remain undecidable, including Hamiltonicity and ExactCover [23,24]. Some problems that are polynomial-time decidable in the finite case, including 2-Sat, 2-colorability [23], and checking whether a graph is connected [3], are undecidable on automatic structures.

Another restricted class is that of *doubly periodic structures* [8]. These are constructed by placing infinitely many copies of a fixed finite structure on a two-dimensional grid, and imposing additional relations on nodes from neighbouring copies only, in a periodic manner. On this class, k-Sat and k-Colorability are decidable for $k = 2$ and undecidable for $k > 2$ [9], and one could hope that this could be a setting where the P vs. NP gap is blown up to the gap between decidable and undecidable problems. However, all doubly periodic structures are automatic, so decidability results mentioned above hold here as well.

As we said above, our focus is on structures which are first-order interpretable in a pure set. In the following we will simply call such structures *definable*. All such structures are automatic, since the class of automatic structures is closed under first-order interpretations [2]. They are incomparable with doubly periodic ones. (For example, an infinite clique is definable but not doubly periodic, and an infinite square grid is doubly periodic but not definable.)

A few classical decision problems have been studied in the framework of definable structures. In [19], it was proved that every Constraint Satisfaction Problem for a fixed finite template is decidable in this setting. This includes k-Sat and k-colorability, for every k. On the other hand, checking whether there exists a homomorphism from one given definable structure to another is undecidable [20]. For solvability of systems of linear equations, [19] shows decidability over finite fields and where every equation contains finitely many variables. Both these assumptions are dropped in [10], at the price of searching for definable solutions only. In [11], linear programming over definable structures is shown to be decidable, and integer linear programming undecidable, again under the assumption that only definable solutions are considered.

The decidability landscape of Karp's NP-complete problems on definable structures turns out to be surprisingly varied. Out of the original 21 problems:

- Seven are undecidable, including CNF-Sat, ExactCover, and Hamiltonicity. We show this by a sequence of reductions, starting from a reduction of the Wang tiling problem to ExactCover.
- Eleven are decidable. Two of these, 3-Sat and Colorability, are known from [19], and we generalise the technique used there to cover a few more, including VertexCover. Decidability of Clique and SetPacking is known for automatic structures (see above), but for definable structures we provide more direct arguments.
- Three problems essentially rely on adding up unboundedly many numbers, and we see no natural way to extend them to an infinite setting.

2 Preliminaries: definable structures

We assume general familiarity with the classic paper [16] and the basic concepts discussed there such as graphs, cliques, formulas etc. We will recall along the way the 21 computational problems studied there. In this preliminary section we focus on *definable structures*, which we will use as instances for those problems. We will work in the framework of *sets with atoms* [4], also called *nominal sets* [26]. Our presentation follows [19,20]; see there and [4, Chap. 10] for more details.

Let $\mathbb{A}$ be a countably infinite set of *atoms*. We want to define sets that are somehow built of atoms and are perhaps infinite, but presented in a finite way and highly symmetric under bijective atom renaming. To this end, given some fixed infinite set of atom variables, an *expression* is either a variable or a formal finite (perhaps empty) union of set-builder expressions of the form

$$\{e \mid v_1, \ldots, v_k \in \mathbb{A},\ \phi\}$$

where e is an expression (where the v_i as well as other variables may occur), the v_i are bound variables, and ϕ is a first-order formula over the set of atom variables, with equality as the only relation symbol. Free variables in this expression are those free variables in e and ϕ which are not among the $v_1, \ldots, v_k$. For an expression e with free variables V, any valuation $\sigma : V \to \mathbb{A}$ defines a value

$X = e[\sigma]$ in an obvious way, by induction on the structure of e. This value is either an atom or a set. We will call X a *definable set with atoms*. If the image of σ is some (necessarily finite, and perhaps empty if e has no free variables) $S \subseteq \mathbb{A}$, then we may say that X is *S-definable*. $\emptyset$-definable sets are called *equivariant*. For example, the equivariant set $\binom{\mathbb{A}}{2}$ from Example 1 is formally defined by the expression

$$\{\{u\} \cup \{v\} \mid u, v \in \mathbb{A}, u \neq v\},$$

where the subexpressions $\{u\}$ and $\{v\}$ use a little syntactic sugar: the empty list of bound variables, as well as a formula ϕ which is always true, can be elided.

The above language of expressions is rudimentary, but using standard set-theoretic machinery it is easy to extend it with pairs, tuples, integers, constants from some given finite sets, atom constants, etc. We will use such syntactic sugar without further warning. In particular, in Examples 1-2, not only the sets X and $\mathcal{S}$ but also the pair $(X, \mathcal{S})$ is definable (in fact, equivariant).

Furthermore (see [4, Chap. 10] for a detailed discussion), it is routine to prove by induction on the structure of expressions that definable sets are closed under Boolean combinations, Cartesian products, images and inverse images of definable functions, quotients under definable equivalence relations, and intersections and unions of definable families, and all these constructions are effectively computable as operations on the defining expressions. Also relations such as set equality and membership are decidable. As a result, set-builder expressions can be safely extended with more syntactic sugar by allowing bound variables to range not only over $\mathbb{A}$ but over any definable set, and allowing in ϕ set relations $\in$ and $\subseteq$ (in addition to atom equality) and quantifiers of the form $\exists x \in X$ and $\forall x \in X$, where X is a set defined by an expression. For example, for $(X, \mathcal{S})$ as in Examples 1-2, the sets

$$\{(x, Y) \mid x \in X, Y \in \mathcal{S}, x \in Y\} \qquad \text{and} \qquad \{\{Y \in \mathcal{S} \mid x \in Y\} \mid x \in X\}$$

are definable (and equivariant).

Definable structures are highly symmetric. The group $\mathrm{Aut}(\mathbb{A})$ of *atom automorphisms* (meaning: arbitrary bijections on $\mathbb{A}$) has a canonical action on the class of definable sets: for a definable set $X = e[\sigma]$ and $\pi \in \mathrm{Aut}(\mathbb{A})$, define $X \cdot \pi = e[\sigma; \pi]$, where $\sigma; \pi$ denotes function composition: $(\sigma; \pi)(v) = \pi(\sigma(v))$. This amounts to consistently renaming all the atoms throughout X according to π. For a finite $S \subseteq \mathbb{A}$, a $\pi \in \mathrm{Aut}(\mathbb{A})$ is called an *S-automorphism* if $\pi(a) = a$ for all $a \in S$. We say that S *supports* a definable set X if $X \cdot \pi = X$ for every S-automorphism π. It is easy to see that every S-definable set is supported by S, so every definable set has a finite support. Finite supports of a given set are closed under intersection (see [4, Thm. 4.13] or [26, Prop. 2.3]), so every definable set X has a *least support*, denoted $\mathrm{supp}(X)$.

We say that sets X and Y are S-equivalent if there is an S-automorphism π such that $X \cdot \pi = Y$. This is an equivalence relation, and its equivalence classes are called *S-orbits*, or simply *orbits* if $S = \emptyset$. Every definable set X is *orbit-finite*, i.e., it is a finite union of $\mathrm{supp}(X)$-orbits. For example:

- $\mathbb{A}$ is a single-orbit set. For any finite $S \subseteq \mathbb{A}$, the set S has $|S|$ S-orbits (every element of S is a singleton orbit), and $\mathbb{A} \setminus S$ has one S-orbit.
- For any k, the sets $\mathbb{A}^{(k)}$ (of non-repeating k-tuples) and $\binom{\mathbb{A}}{k}$ (of sets of size k) are single-orbit sets. More generally, for any finite permutation group $G \leq \mathrm{Sym}(k)$, the set $\mathbb{A}^{(k)}/G$ of non-repeating k-tuples of atoms up to permutations from G, is a single-orbit set.
- The set $\mathbb{A}^2$ (of possibly repeating pairs) consists of two orbits; more generally $\mathbb{A}^k$ has number of orbits equal to the k-th Bell number.
- The set of finite subsets of $\mathbb{A}$ is not orbit-finite, as sets of different sizes fall into different orbits; therefore this set is not definable.

There is an alternative but equivalent way to introduce definable sets, where actions of $\mathrm{Aut}(\mathbb{A})$ and finite supports are the basic concepts, with orbit-finiteness imposed as an additional condition. It then becomes a representation theorem that every S-supported orbit-finite set is (in an S-suppported bijection with) an S-definable set. This approach is taken in [4,5,6]. Other representations exist: as shown in [6], every equivariant single-orbit set is in equivariant bijection with a set of the form $\mathbb{A}^{(k)}/G$ as mentioned above. This implies that definable structures are first-order interpretable (in the sense of model theory) in the pure set $(\mathbb{A}, =)$.

So far we have focused on *equality atoms*, where $\mathbb{A}$ is a pure set, without any structure imposed on the atoms. This is our main subject of study here, and in the following we will study Karp's problems only on structures definable over equality atoms. However, in Section 4, as in [19], we will need to make a brief excursion to a richer structure of *ordered atoms*, where $\mathbb{A} = (\mathbb{Q}, \leq)$ is the total order of rationals, with $\mathrm{Aut}(\mathbb{A})$ restricted to order-preserving bijections. This extends the language of formulas ϕ in set-builder expressions with a binary order relation $\leq$. The notions of definability, support and orbit-finiteness are defined as for equality atoms, and the representation theorems mentioned above work analogously. The general framework of sets over any relational structure of atoms is studied in detail in [4,6].

3 Undecidable problems

We will list Karp's problems that become undecidable over definable structures.

3.1 Exact cover

Problem 1 (ExactCover).
Input: Definable set X, definable set $\mathcal{S}$ of subsets of X
Question: Is there a pairwise-disjoint subset of $\mathcal{S}$ whose union is X?

In the classical setting, where both X and $\mathcal{S}$ are finite, Karp [16] proved NP-hardness of this problem by reduction from graph colorability. Looking for a similar reduction here would be pointless, since colorability of definable graphs is decidable. Instead, we prove undecidability by a reduction from the well-known domino tiling problem, posed by Wang [28] and shown undecidable by Berger [1].

To describe that problem, fix constants N, S, E and W. For a finite set C of colors, a *tile* is a function $t : \{\mathsf{N}, \mathsf{S}, \mathsf{E}, \mathsf{W}\} \to C$, and for a set T of tiles, a *tiling* of the plane is a function $\tau : \mathbb{Z}^2 \to T$ such that, for all $a, b \in \mathbb{Z}$:

$$\tau(a,b)(\mathsf{N}) = \tau(a, b+1)(\mathsf{S}) \qquad \text{and} \qquad \tau(a,b)(\mathsf{E}) = \tau(a+1, b)(\mathsf{W}).$$

*Problem 2 (*DOMINO*).*

Input: A finite set C of colors, a finite set T of tiles over C
Question: Is there a tiling with T?

We will reduce DOMINO to EXACTCOVER. Given an input (C, T), we will construct definable X and $\mathcal{S}$ such that an exact cover of $(X, \mathcal{S})$ exists if and only if a tiling of the plane does.

For some intuition, imagine an infinite clique with all atoms as vertices. The set X will contain 4 elements for each atom (the *vertex elements*), and a few elements for each pair of distinct atoms (the *edge elements*). The family $\mathcal{S}$ will contain two kinds of (finite) sets. An *edge set* will contain all the edge elements associated to a specific pair. A *tile set*, defined over a quadruple of distinct atoms a, b, c, d, will contain some of the vertex elements associated to a, b, c and d, as well as some edge elements associated to the pairs $(a, b), (b, c), (c, d)$ and (d, a). We can consider it associated with the quadrilateral a, b, c, d.

Only tile sets contain vertex elements, so an exact cover in $\mathcal{S}$ must use infinitely many of them. Each tile set will only partially fill its four edges (i.e. it will not include an entire edge set), so in an exact cover each tile set must match other tile sets complementing it on its edges, in a way that can be unrolled to a tiling of the infinite square grid. Edge sets are used to cover all the edge elements that do not correspond to edges in that grid.

We shall now make these ideas precise.

Theorem 1. *EXACTCOVER is undecidable.*

Proof. By reduction from DOMINO. Given as input finite sets C of colors and T of tiles, we shall construct an instance $(X, \mathcal{S})$ of EXACTCOVER. Define:

$$X = \bigcup_{a \in \mathbb{A}} \{\mathsf{NW}_a, \mathsf{NE}_a, \mathsf{SE}_a, \mathsf{SW}_a\} \quad \cup \bigcup_{(a,b) \in \mathbb{A}^{(2)}} \{\rightarrow_{(a,b)}\}$$
$$\cup \bigcup_{\{a,b\} \in \binom{\mathbb{A}}{2}} \{1_{\{a,b\}}, 2_{\{a,b\}}, 3_{\{a,b\}}, 4_{\{a,b\}}\} \cup \{c_{\{a,b\}} : c \in C\}.$$

Note that X depends on C but not on T. For $\{a, b\} \in \binom{\mathbb{A}}{2}$, define the *edge set* $e_{\{a,b\}}$ by:

$$e_{\{a,b\}} = \{1_{\{a,b\}}, 2_{\{a,b\}}, 3_{\{a,b\}}, 4_{\{a,b\}}, \rightarrow_{(a,b)}, \rightarrow_{(b,a)}\} \cup \{c_{\{a,b\}} : c \in C\}.$$

For a tile $t \in T$ and atoms $(a, b, c, d) \in \mathbb{A}^{(4)}$, define the *tile set* $t_{(a,b,c,d)}$ by:

$$\begin{aligned} t_{(a,b,c,d)} = \{&\mathsf{SE}_a, \mathsf{SW}_b, \mathsf{NW}_c, \mathsf{NE}_d, \rightarrow_{(a,b)}, \rightarrow_{(b,c)}, \rightarrow_{(c,d)}, \rightarrow_{(d,a)}, \\ &1_{\{a,b\}}, 2_{\{a,b\}}, 1_{\{b,c\}}, 3_{\{b,c\}}, 3_{\{c,d\}}, 4_{\{c,d\}}, 2_{\{d,a\}}, 4_{\{d,a\}}\} \\ &\cup \{c_{\{a,b\}} : c \in C \wedge t(\mathsf{N}) = c\} \cup \{c_{\{b,c\}} : c \in C \wedge t(\mathsf{E}) = c\} \\ &\cup \{c_{\{c,d\}} : c \in C \wedge t(\mathsf{S}) \neq c\} \cup \{c_{\{d,a\}} : c \in C \wedge t(\mathsf{W}) \neq c\} \end{aligned}$$

(see Figure 1 for intuition). Let $\mathcal{S}$ contain all the edge and tile sets as above.

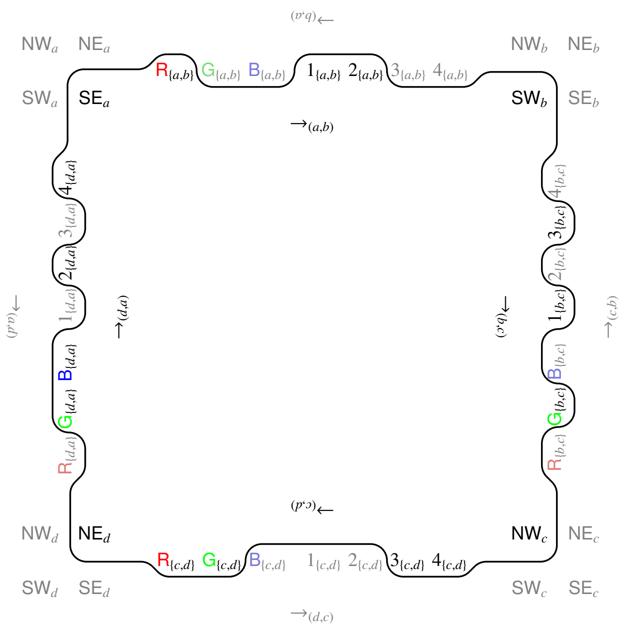


Fig. 1: The set $t_{(a,b,c,d)}$ for $C = \{\mathsf{R}, \mathsf{G}, \mathsf{B}\}$, $t = (\mathsf{N} \mapsto \mathsf{R}, \mathsf{E} \mapsto \mathsf{G}, \mathsf{S} \mapsto \mathsf{B}, \mathsf{W} \mapsto \mathsf{R})$, with the excluded neighbouring elements marked as gray

We shall show that if $(X, \mathcal{S})$ admits an exact cover then T tiles the plane.

Fix an exact cover $E \subseteq \mathcal{S}$. Let F be the set of tile sets in E, and note it is infinite (because edge sets don't contain elements associated with atoms, only pairs of atoms, and tile sets are finite). Consider a tile set $t_{(a,b,c,d)} \in F$. It covers $\rightarrow_{(a,b)}$ but not $\rightarrow_{(b,a)}$, so there is a unique tile set $n(t_{(a,b,c,d)})$ covering its complement on the elements associated to both of a and b. We can similarly define $e(t_{(a,b,c,d)}), s(t_{(a,b,c,d)})$ and $w(t_{(a,b,c,d)})$ by looking at elements associated respectively to b, c, to c, d, and to d, a. Doing this for all tile sets in F defines functions $n, e, s, w : F \to F$.

Moreover, for any $t_{(a,b,c,d)} \in F$ we have $n(t_{(a,b,c,d)}) = t'_{(e,f,b,a)}$ for some $t' \in T$ and $e, f \in \mathbb{A}$ (otherwise $t_{(a,b,c,d)}$ and $n(t_{(a,b,c,d)})$ intersect on at least one of $1_{\{a,b\}}, 2_{\{a,b\}}, 3_{\{a,b\}}, 4_{\{a,b\}}, \rightarrow_{(a,b)})$. Hence $s \circ n = \mathrm{id}$, and similar arguments show $n \circ s = w \circ e = e \circ w = \mathrm{id}$. Furthermore, $e(n(t_{(a,b,c,d)}))$ and $n(e(t_{(a,b,c,d)}))$ both contain NE_b, so they are equal and $n \circ e = e \circ n$. Similarly $n \circ w = w \circ n$, $s \circ e = e \circ s$ and $s \circ w = w \circ s$.

Lastly, note that if $n(t_{(a,b,c,d)}) = t'_{(e,f,b,a)}$ then $t(\mathsf{N}) = t'(\mathsf{S})$ (else the tile sets intersect on $c_{\{a,b\}}$ for some $c \in C$) and similarly in the east-west direction.

But then take any $t_{(a,b,c,d)} \in F$ and define $\phi : \mathbb{Z}^2 \to F$ by $\phi(0,0) = t_{(a,b,c,d)}$ and $\phi(x, y+1) = n(\phi(x,y))$, $\phi(x+1, y) = e(\phi(x,y))$ for all $x, y \in \mathbb{Z}$. Composing with the map taking a tile set to its associated tile, we obtain a tiling of $\mathbb{Z}^2$. Note that ϕ may not be injective, but this is not a problem. It only implies that the encoded tiling may be periodic.

Next we must show that a tiling gives rise to an exact cover. Fix a tiling $\tau : \mathbb{Z}^2 \to T$. Choose a bijection $\alpha : \mathbb{Z}^2 \to \mathbb{A}$ and (with $\|\cdot\|$ the Euclidean norm on $\mathbb{Z}^2$) define:

$$E = \left\{(\tau(x,y))_{(\alpha(x,y+1),\alpha(x+1,y+1),\alpha(x+1,y),\alpha(x,y))} : (x,y) \in \mathbb{Z}^2\right\} \\ \cup \left\{e_{\{a,b\}} : a, b \in \mathbb{A} \wedge \|\tau^{-1}(b) - \tau^{-1}(a)\| > 1\right\}.$$

Intuitively, α arranges the atoms into a grid. For every unit square in the grid, E contains the tile set corresponding to $\tau(x,y)$ and to the atoms associated to the four corners of the square. This covers (with pairwise disjoint sets) all elements associated to atoms or to those pairs of atoms which τ maps to adjacent points of the grid (i.e. of distance 1). The remaining elements of X are all associated with pairs of atoms not (in this sense) adjacent. We cover these with edge sets.

Finally, X and $\mathcal{S}$ are definable and the construction is effective, so the reduction is complete. □

Remark 1. In our construction, all sets in $\mathcal{S}$ are finite. Hence ExactCover under this restriction is undecidable as well. More can be said: a finite set of bounded size can be replaced, in a definable way, by all possible linear orderings of it. As a result, the following variant of the problem, where $\mathcal{S}$ consists of non-repeating tuples rather than subsets, remains undecidable:

Problem 3 (ExactTupleCover).
Input: Definable set X, number n, definable set $\mathcal{S} \subseteq X^{(n)}$
Question: Is there $\mathcal{S}' \subseteq \mathcal{S}$ where every $x \in X$ occurs in exactly one tuple?

We will use this variant in Theorems 5 and 7.

3.2 Hitting set, Satisfiability, 0-1 Integer Linear Programming

A few problems allow straightforward reductions from ExactCover.

Problem 4 (HittingSet).
Input: Definable set P, definable set $\mathcal{F}$ of subsets of P
Question: Is there a subset $Q \subseteq P$ such that $|Q \cap C| = 1$ for every $C \in \mathcal{F}$?

In the finite setting [16], Karp demonstrated NP-hardness of this problem, noticing that it is essentially ExactCover in disguise. The same reduction works here.

Theorem 2. *HittingSet is undecidable.*

Proof. Given an instance $(X, \mathcal{S})$ of ExactCover, define $P = \mathcal{S}$ and

$$\mathcal{F} = \{\{Y \in \mathcal{S} \mid x \in Y\} \mid x \in X\}.$$

Then exact covers for $(X, \mathcal{S})$ are exactly hitting sets for $(P, \mathcal{F})$. □

A CNF propositional formula over a set X of variables can be represented as a family of subsets of $X \times \{0, 1\}$, each of the subsets representing a disjunctive clause. We say that the formula is definable if the family is definable. The notion of a satisfying assignment is as in the finite case.

Problem 5 (CNFSat).
Input: Definable CNF formula ϕ; **Question:** Is ϕ satisfiable?

Theorem 3. *CNFSat is undecidable.*

Proof. By reduction from HittingSet. Given an instance $(P, \mathcal{F})$, write the following formula over variables from P:

$$\left(\bigwedge_{C\in\mathcal{F}} \bigvee_{p\in C} p\right) \wedge \left(\bigwedge_{C\in\mathcal{F}} \bigwedge_{p\neq q\in C} (\neg p \vee \neg q)\right).$$

Its satisfying assignments correspond to hitting sets for $\mathcal{F}$. □

There is an alternative proof of undecidability for CNFSat that avoids our reduction from Theorem 1, and proceeds directly by a straightforward reduction from the satisfiability problem for the ∀∃-fragment of first order logic [7]. [1] The reduction from Theorem 3 is still worthwhile though. Later, in Theorem 9, we will show that 3-Sat, and more generally CNFSat where all clauses are finite, is decidable. Our reduction then implies that, as a counterpoint to Remark 1, the following problems are decidable:

– HittingSet for instances $(P, \mathcal{F})$ where all sets in $\mathcal{F}$ are finite,
– ExactCover for instances $(X, \mathcal{S})$ where every $x \in X$ belongs to finitely many sets in S.

One could try to use the reduction from ∀∃-FO to dispense with our reduction from Theorem 1 altogether. Indeed, there is an easy reduction from CNFSat to HittingSet (and thus to ExactCover, by reversing the construction from Theorem 2): given a formula ϕ, begin by creating a HittingSet instance over

[1] We are grateful to M. Bojańczyk for pointing this out. We do not give the details, not to deprive the reader of the pleasure of solving the corresponding exercise in Section 3.3 of [4].

the variables and their formal negations, and add a set $\{v, \neg v\}$ for each v. It then suffices to add, for each clause C of ϕ, extra elements and sets which force at least one literal in C to be included in any hitting set. This is easy to do: the gadget shown in Figure 2 (where elements are shown as dots and sets are circled) ensures that at least one of the blue elements is included in every hitting set.

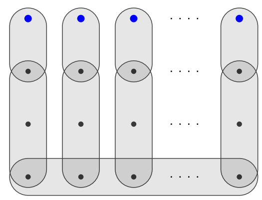


Fig. 2: A HittingSet instance forcing the inclusion of a blue element

This proves that HittingSet, and by extension ExactCover, are undecidable. However, this construction would not let us show undecidability of ExactTupleCover (or ExactCover on instances $(X, \mathcal{S})$ where all sets in $\mathcal{S}$ are finite). Our construction from Theorem 1 does prove that (see Remark 1), and it will be an essential starting point of further reductions in Theorems 5 and 7.

We now turn attention to 0-1 Integer Linear Programming. For a definable set X of variables, an equation is a pair (c, n), where $c : X \to \mathbb{Z}$ is a definable function and $n \in \mathbb{Z}$. This is intended to represent the equation $\sum_{x \in X} c_x x = n$. A valuation $v : X \to \{0, 1\}$ solves the equation if (i) $v(x)$ and c_x are simultaneously non-zero for only finitely many x's, and (ii) $\sum_{x \in X} c_x v(x)$ equals n. (The first condition makes the sum well defined.) The problem is then posed as:

Problem 6 (0-1-ILP).
Input: Definable set X, definable set E of equations over X
Question: Does E have a solution?

Theorem 4. *0-1-ILP is undecidable.*

Proof. Given an instance $(P, \mathcal{F})$ of HittingSet, put $X = P$ and for each $C \in \mathcal{F}$ add the equation $\sum_{p \in C} p = 1$ to E. Solutions to E are hitting sets for $\mathcal{F}$. □

This result is not entirely new. In [11, Sec. 9], undecidability is proved for Integer Linear Programming but without the 0-1 restriction, with inequalities allowed in addition to equations, and with the additional condition that only finite solutions are sought. However, an inspection of that proof shows that only 0-1 variables are actually used and inequalities can be encoded as equations. Moreover, the proof can be adapted to deal with arbitrary solutions.[2]

[2] We are grateful to A. Ghosh, P. Hofman and S. Lasota for a helpful discussion about this issue.

3.3 Hamiltonicity

In the finite setting [16], Karp considered the existence of a Hamiltonian cycle, both in directed and undirected graphs. It is not clear what an infinite cycle would mean, so in the definable setting we study paths which are doubly-infinite, i.e. extending infinitely in both directions. The problem makes sense both for directed and undirected graphs:

Problem 7 ((UN)DIRECTEDHAMILTONICITY).
Input: Definable (un)directed graph G
Question: Does G have a doubly-infinite Hamiltonian path?

Theorem 5. *DIRECTEDHAMILTONICITY is undecidable.*

Proof. By a reduction from EXACTTUPLECOVER. Take as input an instance $(X, \mathcal{S})$ (with $\mathcal{S} \subseteq X^{(n)}$ a set of non-repeating tuples). We may assume that $\mathcal{S}$ is infinite. Let us construct $G = (V, E)$ which has a doubly-infinite Hamiltonian path if and only if $(X, \mathcal{S})$ admits an exact cover.

We intend G to consist of a big independent set of vertices corresponding to elements of X, together with a gadget for each $\overline{x} \in \mathcal{S}$. The gadgets will mediate how a doubly-infinite Hamiltonian path can visit vertices corresponding to elements of X. Formally, define:

$$V = X \uplus \{(\overline{x}, i) \mid \overline{x} \in \mathcal{S},\ 1 \leq i \leq 3n+3\}.$$

The vertices $(\overline{x}, i)$ will form the gadget corresponding to $\overline{x}$; the first and the last of them will be called the *ends* of the gadget.

For edges, first connect vertices in each gadget as illustrated in Figure 3; then connect all ends of all gadgets, joining them into a single infinite clique.

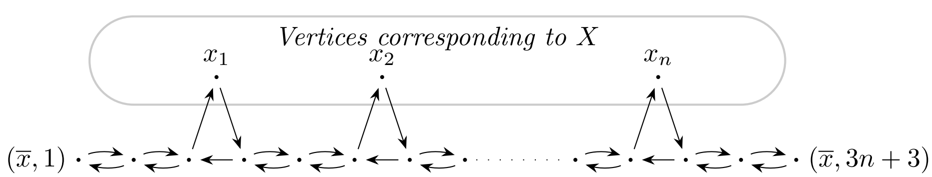
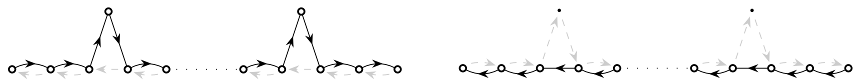


Fig. 3: The gadget corresponding to $\overline{x} = (x_1, \ldots, x_n)$

Consider a doubly-infinite Hamiltonian path in this graph. Note it can visit the gadget corresponding to $(x_1, \ldots, x_n)$ in two ways: from first vertex to last, passing through all of $x_1, \ldots, x_n$, or from last to first, visiting no other vertex along the way (see Figure 4). Hence every $x \in X$ has a unique gadget joined to it which is visited first-to-last. The corresponding tuples form an exact cover.

Conversely, fix an exact cover E. $\mathcal{S}$ is infinite, so there are infinitely many gadgets. Consider a doubly-infinite path which visits gadgets corresponding to E from first to last, and others from last to first. Correctness is clear. □



(a) From first vertex to last (b) From last vertex to first

Fig. 4: The two ways to visit a gadget

There is an easy reduction from directed to undirected Hamiltonicity in the classical case [16], and it goes through in the definable setting with no change.

Theorem 6. *UndirectedHamiltonicity is undecidable.*

Proof. By reduction from DirectedHamiltonicity. Given a definable digraph $G = (V, E)$, form an undirected graph G' with three vertices v_{in}, v_{mid}, v_{out} for each $v \in V$. For each $v \in V$ add edges $\{v_{in}, v_{mid}\}, \{v_{mid}, v_{out}\}$, and for each $(u, v) \in E$ add an edge $\{u_{out}, v_{in}\}$. Then a Hamiltonian path in G' must - up to reversal - always visit v_{out} after v_{mid} after v_{in}. Hence these correspond to Hamiltonian paths in G. □

It is not difficult to modify our constructions to work for *singly*-infinite Hamiltonian paths, which have a starting point and extend to infinity in one direction. The corresponding problems remain undecidable.

3.4 3D matching

3D matching is a generalisation of bipartite matching to 3-hypergraphs. Some classical formulations of this problem do not adapt to an infinite setting well, but the following one does.

Problem 8 (3DMatching).
Input: Definable sets A, B, C, definable set $R \subseteq A \times B \times C$
Question: Is there a subset of R whose projections onto A, B, C are bijections?

Karp in [16] proved hardness by a reduction from ExactCover. That reduction does not quite work in the definable setting, but we give one that works, based on the same general idea.

Theorem 7. *3DMatching is undecidable.*

Proof. By reduction from ExactTupleCover. Given an instance $(X, \mathcal{S})$ (with $\mathcal{S} \subseteq X^{(n)}$ a set of non-repeating tuples), define:

$$Y = \{(\overline{x}, i) \mid \overline{x} \in \mathcal{S}, 0 \le i < n\}, \qquad A = Y \uplus \mathbb{A}, \qquad B = X \uplus \mathbb{A}.$$

Then let $R \subseteq A \times A \times B$ contain three types of triples:

(i) $((\overline{x}, i), (\overline{x}, i), x_i)$, where $(\overline{x}, i) \in Y$ for $\overline{x} = (x_0, \ldots, x_{n-1})$;
(ii) $((\overline{x}, i), (\overline{x}, i+1), a)$, where $(\overline{x}, i) \in Y$, $a \in \mathbb{A}$, and the $+1$ is modulo n,
(iii) all triples from $\mathbb{A}^3$.

In a 3D matching for R, each $(\overline{x}, i)$ (from either of the first two components) can be covered by a triple of type (i) or of type (ii). The type (i) triple covers x_i in the third component as well, while the type (ii) triple does not. Importantly, if $(\overline{x}, i)$ is covered by a type (ii) triple, then all $(\overline{x}, j)$ in both components must be covered by type (ii) triples too. Hence for each $x \in X$ there is exactly one tuple in $\mathcal{S}$ which contains x and is associated to elements $(\overline{x}, i)$ covered by triples of type (i). These tuples form an exact cover.

Conversely, take an exact cover $E \subseteq S$. For each $\overline{x} \in E$, include all the triples of type (i) associated to it in the matching. For all other tuples, include a minimal set of triples of type (ii), each taking a fresh atom as the third component. We may assume there are infinitely many atoms left unused in B; choose any bijection σ with all of $\mathbb{A}$ and include $(\sigma a, \sigma a, a) \in \mathbb{A}^3$ for each such unused a. □

4 Decidability from extreme amenability

We now turn attention to those problems from Karp's list that remain decidable in the definable setting. First we mildly generalise the technique used in [19], used there to prove that finite-template Constraint Satisfaction Problems are decidable. This generalisation will let us prove decidability for a whole range of Karp's problems in one stroke. The few remaining problems, which are not covered by the technique, will be dealt with in Sections 5-6.

4.1 Compact sets of structures

In this section we need to consider both equality and ordered atoms. The following lemmas apply to both these structures (indeed to all oligomorphic [15] atom structures).

Fix a finite relational signature σ and a definable, equivariant set X. The set Σ of all σ-structures on X comes equipped with a topology, with a basis of neighbourhoods of the form:

$$\mathcal{B}_Y(M) = \{N \in \Sigma \mid N|_Y = M|_Y\}$$

for $Y \subseteq X$ finite and $M \in \Sigma$, where $M|_Y$ is the induced substructure of M.

Lemma 1. *The space Σ is compact, and the canonical action of* $\mathrm{Aut}(\mathbb{A})$ *on Σ is continuous.*

Proof. The set Σ is in bijection with $\prod_{R\in\sigma}\{0,1\}^{X^{\mathrm{arity}(R)}}$, and if σ is finite then our topology is the product topology on this set, which is compact by Tychonoff's theorem. The action is continuous since $\mathcal{B}_Y(M)\cdot\pi = \mathcal{B}_{Y\cdot\pi}(M\cdot\pi)$ for every π. □

For a structure $\mathcal{M} \in \Sigma$ we write $\mathrm{Age}_{\mathbb{A}}(M)$ to denote the $\mathrm{Aut}(\mathbb{A})$-closure of the set of finite induced substructures of M, and define a preorder $\sqsubseteq$ on Σ by

$$M \sqsubseteq N \iff \mathrm{Age}_{\mathbb{A}}(M) \subseteq \mathrm{Age}_{\mathbb{A}}(N).$$

For a structure M, the down-set $M{\downarrow}$ is the set of all N such that $N \sqsubseteq M$.

Lemma 2. *For every $M \in \Sigma$, the space $M{\downarrow} \subseteq \Sigma$ is equivariant and compact.*

Proof. Equivariance is obvious: if M, N are in the same $\mathrm{Aut}(\mathbb{A})$-orbit of Σ then they are $\sqsubseteq$-equivalent. For compactness, by Lemma 1 it is enough to show that $M{\downarrow}$ is closed. Take any $N \not\sqsubseteq M$. Then for some finite Y the structure $N|_Y$ is not in $\mathrm{Age}_{\mathbb{A}}(M)$ so $N|_Y \neq M|_Y$ and the open $\mathcal{B}_Y(N)$ is disjoint from $M{\downarrow}$. □

The following result would apply to any atoms with the Ramsey property [17], but we will only need it for ordered atoms with finitely many pairwise distinct constants, so this is how we formulate it.

Theorem 8. *Over atoms $\mathbb{A} = (\mathbb{Q}, \leq, a_1, \ldots, a_n)$, for any X and σ as above, every down-set in Σ contains an equivariant structure.*

Proof. Consider any down-set $M{\downarrow} \subseteq \Sigma$. By Lemma 2 it is equivariant, so the continuous action from Lemma 1 restricts to it. By Pestov's theorem [25], the group $\mathrm{Aut}(\mathbb{Q}, \leq)$ is *extremely amenable*: every continuous action of it on a nonempty compact space has a fixpoint. Fixpoints of the canonical action of $\mathrm{Aut}(\mathbb{A})$ are exactly equivariant elements. This proves the theorem for $n = 0$. For the general statement, note that $\mathrm{Aut}(\mathbb{A})$ is extremely amenable. This follows from the isomorphism $\mathrm{Aut}(\mathbb{A}) \cong (\mathrm{Aut}(\mathbb{Q}, \leq))^{(n+1)}$, since the class of extremely amenable groups is closed under direct products (see [17, Lem. 6.7]). □

4.2 Decidability

Our decidability proofs in this section will follow the pattern used in [19]. Given a problem instance S-definable over equality atoms, we:

(i) See it as an equivariant structure over ordered atoms with constants from S;
(ii) Understand solutions to that instance as structures over a finite relational signature;
(iii) Show that if the set of legal solutions is non-empty then it contains a down-set in the sense of Sec. 4.1 (this is true e.g. if the set of legal solutions is downwards-closed with respect to $\sqsubseteq$);
(iv) Infer from Theorem 8 that if a solution exists then an equivariant one exists;
(v) Show that looking for an equivariant solution is a decidable problem.

As a first example, consider the restriction of CNFSat to formulas with finite clauses. The decidability of this problem follows from [19, Thm. 3], but we reprove it here as an illustrative application of Thm. 8.

Problem 9 (3-Sat).
Input: Definable CNF formula ϕ with at most 3 literals per clause
Question: Is ϕ satisfiable?

Theorem 9. *3-Sat is decidable.*

Proof. Take a 3-CNF formula ϕ, S-definable over equality atoms, over an S-definable set X of variables, for $S \subseteq \mathbb{A}$. (i) If we impose on $\mathbb{A}$ any ordering isomorphic to the total order of the rationals, ϕ remains S-definable over ordered atoms. Adding the atoms from S to $\mathbb{A}$ as constants, ϕ becomes equivariant.

(ii) A valuation for ϕ (satisfying or not) can be seen as a predicate on X, i.e. a structure over a signature σ with a single predicate symbol. (iii) The set of satisfying valuations is described by an equivariant set of forbidden finite substructures, namely the partial valuations which invalidate one of the clauses of ϕ. As a result, the set of satisfying valuations for ϕ is downwards-closed, so (iv) by Theorem 8 it either is empty or contains an equivariant valuation.

To check if ϕ is satisfiable, it is therefore enough to look for a satisfying valuation which is equivariant over our extended atoms or, equivalently, S-definable over ordered atoms. (v) This is easy to do: there are only finitely many S-orbits of variables in X, and an S-definable valuation must be constant in every orbit, so there are finitely many valuations to consider, and for each of them it is easy to decide whether it satisfies every clause in ϕ. □

The same argument shows that k-SAT is decidable for any k. It also means that CNFSAT restricted to formulas with finite clauses, is decidable. This is because every such formula ϕ, having an orbit-finite set of clauses, has an upper bound on the size of a clause, so it is an instance of k-SAT for some k which can be computed from ϕ.

It may be illustrative to see where the argument fails for the unrestricted CNFSAT. Consider an equivariant CNF-formula with two clauses over $X = \{x_a \mid a \in \mathbb{A}\}$:

$$\phi = \left(\bigvee_{a\in\mathbb{A}} x_a\right) \wedge \left(\bigvee_{a\in\mathbb{A}} \neg x_a\right).$$

Consider any satisfying valuation, understood as a structure M on X over a single predicate symbol. Assume that in the valuation infinitely many variables are false. (This does not lose generality: the symmetric case is that infinitely many variables are *true*.) Then $\mathrm{Age}_{\mathbb{A}}(M)$ contains (among other things) all finite substructures where no element satisfies the predicate. Let N be a structure on X where no element satisfies the predicate. Then $N \sqsubseteq M$. But the valuation corresponding to N does not satisfy ϕ, so the set of satisfying valuations does not contain $M{\downarrow}$ and Theorem 8 does not apply. Indeed, even though ϕ has plenty of satisfying valuations, none are equivariant.

The same machinery directly applies to a few more problems. In all these, the input is a definable graph G.

Problem 10 (VERTEXCOVER).
Question: Does G have a finite vertex cover?

Problem 11 (FEEDBACKVERTEXSET).
Question: Is there a finite set of vertices whose removal makes G acyclic?

Problem 12 (FEEDBACKARCSET).
Question: Is there a finite set of edges whose removal makes G acyclic?

In the finite setting of [16], the input to these problems includes a number k, and one asks whether a vertex cover (etc.) of size at most k exists. We could do the same here, but those problems would be decidable for very easy reasons: for a fixed k, there are only orbit-finitely many sets of vertices/edges of size k, and they can be effectively enumerated in the search for a solution. The statements as above are more interesting, but nevertheless:

Theorem 10. *VertexCover, FeedbackVertexSet and FeedbackArcSet are decidable.*

Proof. We proceed as for Thm. 9, arguing only for part (iii). Consider VertexCover. A choice of vertices in G can be seen as a predicate M on the set of all vertices. If M is a finite vertex cover (of size, say, k) then $\mathrm{Age}_{\mathbb{A}}(M)$ can be described by a set of forbidden finite structures: (a) those where more than k vertices satisfy the predicate, and (b) those where neither end of some edge satisfies the predicate. This implies that $M{\downarrow}$ contains only finite vertex covers, and Thm. 8 applies.

The arguments for the remaining two problems are similar. □

Problem 13 (Colorability).
Question: Does G have a vertex coloring with finitely many colors?

Problem 14 (CliqueCover).
Question: Is V a disjoint union of finitely many cliques (where $G = (V, E)$)?

These two problems are equivalent (replace G with its complement to reduce one to the other), so let us focus on colorability. As for Probs. 10-12, in the finite setting of [16] the input includes a number k and a k-coloring is sought for. Unlike for Probs. 10-12, that problem remains interesting in our setting. Decidability of k-Colorability follows from [19, Thm. 3], but it is easy to give an argument analogous to Thm. 9. Here, k-colorings can be seen as structures over a signature with k predicate symbols, and the set of legal k-colorings for any fixed G is downwards-closed.

Unrestricted colorability easily follows from this:

Theorem 11. *Colorability (therefore also CliqueCover) is decidable.*

Proof. Every finite coloring is a k-coloring for some k, so if one exists then an equivariant k-coloring exists. An equivariant coloring must be constant on every orbit of vertices, so it cannot use more colors than there are orbits. The number of orbits can be computed from G, and the set of candidate equivariant colorings can be enumerated and checked for legality. □

5 Other decidable cases

The key feature of the problems from Sec. 4 was that, whenever they admit a solution to an S-definable instance, then an S-definable solution also exists. Some problems do not have this property, for example:

Problem 15 (Clique).
Input: Definable graph G; **Question:** Does G contain an infinite clique?

Consider for instance a graph with ordered pairs from $\mathbb{A}^{(2)}$ as vertices, with edges between exactly those vertices that share the first component. This equivariant graph contains an infinite clique, indeed it is a disjoint union of infinite cliques, but none of these cliques are equivariant.

Nevertheless, the problem is decidable. This has already been proved for automatic graphs [24,27], but for definable graphs a more direct argument exists.

Theorem 12. *Clique is decidable.*

Proof. A definable graph G is orbit-finite, so any infinite clique in it must have an infinite intersection with one of the orbits. We may therefore focus on graphs with one orbit of vertices. Further, in an infinite clique, every edge $\{v_1, v_2\}$ can be colored with its orbit. There are finitely many orbits of edges, so by Ramsey's theorem the clique contains an infinite sub-clique where every edge is in the same orbit. It is therefore enough to decide, for a given orbit of edges in G, whether G contains infinitely many vertices whose every pair belongs to that orbit.

Over equality atoms, this is easy to do: the condition holds for the orbit of $\{v_1, v_2\}$ if and only if v_1 and v_2 are in the same S-orbit, where $S = \mathrm{supp}(v_1) \cap \mathrm{supp}(v_2)$. (This condition is easy to decide by looking at v_1 and v_2.)

To see this, first assume that $v_1 \neq v_2$ are in the same S-orbit. Since $v_1 \neq v_2$, this S-orbit is infinite. Moreover, one can find infinitely many vertices $v_1, v_2, v_3 \ldots$ in this S-orbit so that $\mathrm{supp}(v_i) \cap \mathrm{supp}(v_j) = S$ for all $i \neq j$. For equality atoms, this implies that all pairs $\{v_i, v_j\}$ are in the same S-orbit.

In the other direction, assume an infinite set $v_1, v_2, \ldots$ of vertices such that the pairs $\{v_i, v_j\}$ are in the same orbit for all $i \neq j$. The size of $\mathrm{supp}(v_i)$ and $S_{ij} = \mathrm{supp}(v_i) \cap \mathrm{supp}(v_j)$ does not depend on i and j, so by basic combinatorics the intersection S_{ij} itself does not depend on i and j; call this shared intersection S. For any fixed $i \neq j$, since v_i and v_j are in the same orbit, there is an atom automorphims π_{ij} that preserves S set-wise and such that $v_i \cdot \pi_{ij} = v_j$. Keeping i fixed and choosing more than $|S|!$ j's, there are some $j \neq j'$ such that π_{ij} and $\pi_{ij'}$ agree on S. Then the composition of π_{ij}^{-1} and $\pi_{ij'}$ maps v_j to $v_{j'}$ and fixes S pointwise, hence v_j and $v_{j'}$ are in the same S-orbit. ☐

The following was proved decidable for automatic structures in [23] with a direct argument, rather than by an immediate reduction to Clique:

Problem 16 (SetPacking).
Input: Definable set X, definable family $\mathcal{S}$ of subsets of X
Question: Does $\mathcal{S}$ contain an infinite, pairwise disjoint subfamily?

Theorem 13. *SetPacking is decidable.*

Proof. Given X and $\mathcal{S}$, solve Clique for a graph with $\mathcal{S}$ as the set of vertices, and an edge between two sets if and only if they are disjoint. ☐

Here is another problem that does not fall into the scope of Sec. 4:

Problem 17 (SetCovering).
Input: Definable set X, definable family $\mathcal{S}$ of subsets of X
Question: Does $\mathcal{S}$ contain a finite subfamily C whose union is X?

In [24], a variant of this problem where C is required to be co-infinite (i.e. such that $\mathcal{S} \setminus C$ is infinite) rather than finite, was proved decidable for all automatic structures. The technique used there does not seem applicable to our formulation, but at least on definable instances the following easy argument works.

Theorem 14. *SetCovering is decidable.*

Proof. Clearly we can reduce to the case where X is a single infinite orbit. Assume that a finite family $C \subseteq \mathcal{S}$ covers X. Every set in C has some finite support, and the union of these supports for all sets in C is also finite, so there must be some $x \in Y \in C$ such that $\mathrm{supp}(x)$ and $\mathrm{supp}(Y)$ are disjoint modulo $\mathrm{supp}(X, \mathcal{S})$ (by which we mean their intersection lies in $\mathrm{supp}(X, \mathcal{S})$; this is empty if the instance is equivariant).

On the other hand, assume that such $x \in Y \in \mathcal{S}$ exist. Then Y contains *all* $z \in X$ such that $\mathrm{supp}(z)$ and $\mathrm{supp}(Y)$ are disjoint (again, and in the following, this is modulo $\mathrm{supp}(X, \mathcal{S})$). Take $Y_1, Y_2 \ldots, Y_{|\,\mathrm{supp}(x)|+1}$ in the orbit of Y with pairwise disjoint supports. Then every element of X has support disjoint from some of the Y_i, so it belongs to Y_i. Hence all the Y_i's jointly cover X.

Finally, it is easy to effectively search for $x \in Y \in \mathcal{S}$ as above. □

This gives an alternative decidability proof of VertexCover (Prob. 10), via a straightforward reduction used already by Karp [16]. The argument in Thm. 10 is still worth making though, as it exhibits additional structure in the space of vertex covers that is missing in the more general case of set coverings.

6 Weighted problems

A few problems on Karp's list involve adding up sets of numbers, be it weights of graph edges, penalties in job sequencing etc. This poses an obvious difficulty in generalising these problems to infinite structures. One may try to follow the idea that we used in Secs. 4-5 and, rather than comparing various quantities to a fixed input number k, require them to be finite (or infinite). In the process, however, the essence of a weighted problem usually seems to be lost.

Consider, for example, the problem of finding a minimal Steiner tree in a weighted graph. In the finite formulation [16], given an edge-weighted graph G, a subset W of its vertices and a number k, one asks whether G has a tree of total weight at most k that spans all the vertices in W. In an infinite setting, for a "total weight" to make sense, one has to restrict attention to graphs with non-negative weights. Then one could ask whether a spanning tree with a *finite* total weight exists. (The problem with a given bound k is easily decidable for the same reason as Probs. 10-12.) This is a valid question, but the values of weights are lost in it: it is equivalent to asking whether a spanning tree exists that uses only finitely many edges with non-zero weights. The problem then simplifies to:

Problem 18 (SteinerTree).

Input: Definable graph $G = (V, E)$, definable subsets $W \subseteq V$ and $F \subseteq E$

Question: Does G have a tree that spans all vertices from W and uses finitely many edges from F?

Theorem 15. *SteinerTree is decidable.*

Proof. One needs to check if (G is connected and) the vertices from W fall into finitely many connected components of the graph $(V, E \setminus F)$. To this end, compute its transitive closure (such fixpoint calculations are effective; see [21] for a general study or [22] for an implementation), restrict to the connected components with vertices from W, and solve CliqueCover (Prob. 14). □

Admittedly, the essence of the original weighted problem is lost to some extent in this formulation. For other problems, it gets worse. Karp's problem of finding a maximal cut in a weighted graph becomes:

Problem 19 (MaxCut).

Input: Definable graph $G = (V, E)$, definable subset $F \subseteq E$ of edges

Question: Is there a subset $W \subseteq V$ such that F contains infinitely many edges between W and $V \setminus W$?

Here even the set E becomes irrelevant. It is easy to see that the condition holds if and only if F is infinite, which is obviously decidable.

The remaining three problems on Karp's list [16] are: knapsack, job sequencing and number partitioning. These seem even less open to infinite generalisations than the two above, so we leave them untreated.

7 Conclusion

We chose to consider Karp's 21 problems simply because they are famous and well-studied. But, in a sense, the choice was principled: by fixing the set in advance, we forced our hand to look at a whole range of possible properties, in the definable setting, of problems which all behave similarly over finite structures. So it is interesting how different these properties turned out to be: some problems (e.g. ExactCover) are undecidable for nontrivial reasons, some (e.g. 3-Sat) are decidable by easy algorithms which are correct for deep reasons, some (e.g. Set-Covering) are decidable by algorithms which are correct for mundane reasons, and some (e.g. Knapsack) do not seem to generalise to the definable setting at all. Some of Karp's reductions (e.g. ExactCover to HittingSet) remain in force, others (e.g. CNFSat to 3-Sat) break down. This paints quite an interesting landscape, and shows that we now have a small arsenal of techniques to analyse simple computational problems in the definable setting.

References

1. Berger, R.: The undecidability of the domino problem. Mem. Amer. Math. Soc. **66**, 72 (1966). https://doi.org/10.1090/memo/0066
2. Blumensath, A., Grädel, E.: Automatic structures. In: Procs. 15th Symposium on Logic in Computer Science. pp. 51–62 (2000). https://doi.org/10.1109/LICS.2000.855755
3. Blumensath, A., Grädel, E.: Finite presentations of infinite structures: Automata and interpretations. Theory of Computing Systems **37**, 641–674 (2004). https://doi.org/10.1007/s00224-004-1133-y
4. Bojańczyk, M.: Slightly infinite sets (2025), https://www.mimuw.edu.pl/~bojan/papers/notes-July3.pdf, draft from July 3, 2025
5. Bojańczyk, M., Fijalkow, J., Klin, B., Moerman, J.: Orbit-finite-dimensional vector spaces and weighted register automata. Theoretics **13** (2024). https://doi.org/10.46298/theoretics.24
6. Bojańczyk, M., Klin, B., Lasota, S.: Automata theory in nominal sets. Logical Methods in Computer Science **10** (2014). https://doi.org/10.2168/LMCS-10(3:4)2014
7. Börger, E., Grädel, E., Gurevich, Y.: The classical decision problem. Springer (1997)
8. Burr, S.A.: Some undecidable problems involving the edge-coloring and vertex-coloring of graphs. Discrete Mathematics **50**, 171–177 (1984). https://doi.org/10.1016/0012-365X(84)90046-3
9. Freedman, M.H.: K-sat on groups and undecidability. In: Procs. 30th ACM Symposium on Theory of Computing. p. 572–576 (1998). https://doi.org/10.1145/276698.276871
10. Ghosh, A., Hofman, P., Lasota, S.: Solvability of orbit-finite systems of linear equations. In: Procs. 37th Symposium on Logic in Computer Science (2022). https://doi.org/10.1145/3531130.3533333
11. Ghosh, A., Hofman, P., Lasota, S.: Orbit-finite linear programming. J. ACM **72**(1) (2025). https://doi.org/10.1145/3703909
12. Grädel, E.: Automatic structures: Twenty years later. In: Procs. 35th Symposium on Logic in Computer Science. p. 21–34 (2020). https://doi.org/10.1145/3373718.3394734
13. Hirst, J.L., Lempp, S.: Infinite versions of some problems from finite complexity theory. Notre Dame J. Formal Logic **37**(4), 545–553 (1996). https://doi.org/10.1305/ndjfl/1040046141
14. Hirst, T., Harel, D.: Taking it to the limit: On infinite variants of NP-complete problems. Journal of Computer and System Sciences **53**(2), 180–193 (1996). https://doi.org/10.1006/jcss.1996.0060
15. Hodges, W.: Model Theory. Cambridge University Press (1993). https://doi.org/10.1017/CBO9780511551574
16. Karp, R.M.: Reducibility among combinatorial problems. In: Procs. Symposium on the Complexity of Computer Computations. pp. 85–103 (1972). https://doi.org/10.1007/978-1-4684-2001-2_9
17. Kechris, A.S., Pestov, V.G., Todorcevic, S.: Fraïssé limits, Ramsey theory, and topological dynamics of automorphism groups. GAFA, Geom. Funct. Anal. **15** (2005). https://doi.org/10.1007/s00039-005-0503-1
18. Khoussainov, B., Nerode, A.: Automatic presentations of structures. In: Procs. International Workshop on Logic and Computational Complexity. pp. 367–392 (1995). https://doi.org/10.1007/3-540-60178-3_93

19. Klin, B., Kopczyński, E., Ochremiak, J., Toruńczyk, S.: Locally finite constraint satisfaction problems. In: Procs. 30th Symposium on Logic in Computer Science. pp. 475–486 (2015). https://doi.org/10.1109/LICS.2015.51
20. Klin, B., Lasota, S., Ochremiak, J., Toruńczyk, S.: Homomorphism problems for first-order definable structures. In: Procs. 36th Conference on Foundations of Software Technology and Theoretical Computer Science. LIPIcs, vol. 65, pp. 14:1–14:15 (2016). https://doi.org/10.4230/LIPIcs.FSTTCS.2016.14
21. Klin, B., Łełyk, M.: Scalar and vectorial μ-calculus with atoms. Logical Methods in Computer Science **15** (2019). https://doi.org/10.23638/LMCS-15(4:5)2019
22. Klin, B., Szynwelski, M.: SMT solving for functional programming over infinite structures. In: Procs. 6th Workshop on Mathematically Structured Functional Programming. Electronic Proceedings in Theoretical Computer Science, vol. 207, pp. 57–75 (2016). https://doi.org/10.4204/EPTCS.207.3
23. Köcher, C.: Analyse der Entscheidbarkeit diverser Probleme in automatischen Graphen. Bachelor's thesis, TU Ilmenau (2014)
24. Kuske, D., Lohrey, M.: Some natural decision problems in automatic graphs. The Journal of Symbolic Logic **75**(2), 678–710 (2010). https://doi.org/10.2178/jsl/1268917499
25. Pestov, V.: On free actions, minimal flows, and a problem by Ellis. Trans. Amer. Math. Soc. **350** (1998). https://doi.org/10.1090/S0002-9947-98-02329-0
26. Pitts, A.M.: Nominal sets: names and symmetry in computer science. Cambridge University Press (2013). https://doi.org/10.1017/CBO9781139084673
27. Rubin, S.: Automata presenting structures: A survey of the finite string case. Bull. Symbolic Logic **14** (2008). https://doi.org/10.2178/bsl/1208442827
28. Wang, H.: Proving theorems by pattern recognition — II. The Bell System Technical Journal **40**, 1–41 (1961). https://doi.org/10.1002/j.1538-7305.1961.tb03975.x