Orbit-Finite-Dimensional Vector Spaces and Weighted Register Automata

Mikołaj Bojańczyk
University of Warsaw
Email: bojan@mimuw.edu.pl

Bartek Klin
University of Warsaw
Email: klin@mimuw.edu.pl

Joshua Moerman
RWTH Aachen
Email: joshua@cs.rwth-aachen.de

***Abstract*—We develop a theory of vector spaces spanned by orbit-finite sets. Using this theory, we give a decision procedure for equivalence of weighted register automata, which are the common generalization of weighted automata and register automata for infinite alphabets. The algorithm runs in exponential time, and in polynomial time for a fixed number of registers. As a special case, we can decide, with the same complexity, language equivalence for unambiguous register automata, which improves previous results in three ways: (a) we allow for order comparisons on atoms, and not just equality; (b) the complexity is exponentially better; and (c) we allow automata with guessing.**

I. Introduction

Weighted automata over a field were introduced in [17] by Schützenberger. Such an automaton is defined in the same way as a nondeterministic automaton, with a set Q of states and an input alphabet Σ, except that instead of having subsets for transitions, initial and final states, the automaton has weight functions into the underlying field:

$$\underbrace{I\colon Q \to \mathbb{F}}_{\text{initial}} \quad \underbrace{\delta\colon Q \times \Sigma \times Q \to \mathbb{F}}_{\text{transition}} \quad \underbrace{F\colon Q \to \mathbb{F}}_{\text{final}}.$$

The weight of a run is obtained by multiplying the weights along all transitions, the initial weight of the first state, and the final weight of the last state. The automaton recognizes a weighted language, which is the function $L\colon \Sigma^* \to \mathbb{F}$ that maps a word to the sum of weights of all runs on that word.

Schützenberger proved that weighted automata can be minimized [17, Sec. B], which provides a polynomial time algorithm for equivalence. In contrast, it is undecidable whether some field element $a \in \mathbb{F}$ is achieved as the weight of some word, already in the special case of weighted automata that are probabilistic [15, Thm. 22]. Equivalence is undecidable for weighted automata over semirings that are not fields, e.g. for the min-plus semiring [12, Cor. 4.3].

One application of weighted automata (see [1] for other ones) is a polynomial time algorithm for language equivalence of unambiguous automata, i.e. nondeterministic finite automata that have at most one accepting run for every word. The algorithm is a simple reduction to equivalence of weighted automata: a nondeterministic automaton can be viewed as a weighted automaton over the field of rational numbers, such that the weight of a word is the number of accepting runs. For unambiguous automata, the number of accepting runs is either zero or one, and hence two unambiguous automata accept the same words if and only if the corresponding weighted automata are equivalent.

In this paper, we generalize weighted automata to infinite alphabets, motivated by the study of register automata, in particular the equivalence problem for unambiguous register automata [7], [8], [13]. The kinds of infinite alphabets that we study are constructed using an infinite set $\mathbb{A}$ of atoms (also called data values) which can only be accessed in very limited ways; in the simplest case, they can only be compared for equality. Register automata are like finite automata, except that they additionally use finitely many registers to store atoms that occurred in the word. This model was introduced by Kaminski and Francez [10, Def. 1], under the name of *finite memory automata*. This model has attracted much attention, and is now one of the most widely studied infinite state systems. The decidability landscape for register automata is rather complex: for example, emptiness is decidable for non-deterministic register automata [10, Thm. 1], but universality is not [16, Thm 5.1]. More robust results can be achieved for deterministic register automata, but at a considerable loss of expressive power.

One of the numerous (unfortunately non-equivalent) variants of register automata are unambiguous register automata, for which language equivalence was shown in [13, Thm. 1] to be decidable in 2ExpSpace, and in ExpSpace for a fixed number of registers. These upper bounds were improved to 2ExpTime and ExpTime, respectively, in [7, Thm. 2]. None of these proofs use weighted automata.

From the point of view of register automata, the main contributions of this paper are:

1) We introduce a weighted version of register automata, and we prove that their equivalence problem can be solved in ExpTime, and in polynomial time for a fixed number of registers.
2) We show that our weighted automata have a robust theory, in particular they can be described in several different ways and admit canonical syntactic automata.
3) As an immediate application of our equivalence algorithm for weighted automata, we show that the language equivalence problem for unambiguous register automata can be solved in the same complexity, improving exponentially on prior work [7], [13].
4) In [7], [13], the equivalence algorithms work only for automata which are non-guessing, in the sense that every

pair (state, input letter) has finitely many outgoing transitions. Without the non-guessing assumption, decidability is only known in the case where one of the automata has just a single register [13, Thm. 10]. Our algorithm can be adapted so that it works for general unambiguous automata, without the non-guessing assumption, with the same complexity.

5) Apart from atoms with equality only, our algorithm for weighted automata, and its applications to unambiguous automata, also work for atoms equipped with a total order. Previous algorithms for ordered atoms assume that one of the automata has a single register [13, Thm. 2].

In our opinion, perhaps the most interesting aspect of this paper is not the applications described above, but the theory that is developed in order to obtain them. Our central objects of study are vector spaces which are spanned by orbit-finite set of vectors. A typical example is the vector space $\mathsf{Lin}(\mathbb{A}^k)$, which consists of finite linear combinations of k-tuples of atoms. Such a vector space has two kinds of structure: we can take linear combinations of vectors, and we can apply atom automorphisms to them. A natural concept – which appears in our equivalence algorithm for weighted automata – is the *equivariant subspaces*, i.e. subsets of the vector space closed under both linear combinations and atom automorphisms. Our principal technical result is that for every k, there is a finite (exponential in k) bound on the maximal length of chains

$$\underbrace{V_1 \subsetneq V_2 \subsetneq \cdots \subsetneq V_n}_{\text{equivariant subspaces}} \subseteq \mathsf{Lin}(\mathbb{A}^k).$$

This bound is used in the equivalence algorithm for weighted automata, but it also has many consequences for vector spaces spanned by orbit-finite sets, e.g. their closure under Cartesian products, tensor products and dual spaces. We hope that these results can be used in future work, beyond applications to unambiguous automata.

II. Orbit-finite sets

Our paper is based on the approach to register automata that uses *sets with atoms*. The idea is to consider automata where all components are closed under atom automorphisms, and the states and input alphabets have finitely many elements up to atom automorphisms. This more abstract view avoids cumbersome notation for representing states and transitions of register automata. We can also leverage existing results that treat atoms with more structure than equality only, e.g. order.

Our notation is based on [3]. Fix a countably infinite relational structure $\mathbb{A}$, in the sense of model theory, i.e. an underlying set equipped with some relations. Elements of this fixed structure will be called *atoms*. In this paper we mostly focus on two such structures:

- the *equality atoms* $\mathbb{A} = (\mathbb{N}, =)$, and
- the *ordered atoms* $\mathbb{A} = (\mathbb{Q}, <)$.

The main results of this paper, notably the finite length property in Section IV, are proved for these atoms only, and we leave other atoms as future work.

An *atom automorphism* is any bijection of the underlying set of atoms which is consistent with all relations in $\mathbb{A}$. For the equality atoms this is any permutation of $\mathbb{N}$; for the ordered atoms, any order-preserving bijection on $\mathbb{Q}$. In both settings, the set of all atoms is orbit-finite, which means that it has finitely many elements up to atom automorphisms. More complicated sets constructed with atoms are also orbit-finite, such as:

$$\underbrace{\mathbb{A}^k}_{k\text{-tuples of atoms}} \qquad \underbrace{\mathbb{A}^{(k)}}_{\substack{\text{non-repeating}\\ k\text{-tuples of atoms}}} \qquad \underbrace{\binom{\mathbb{A}}{k}}_{\text{sets of } k \text{ atoms}}. \tag{1}$$

To formally define "sets constructed with atoms" and the orbit-finite restriction, we use the cumulative hierarchy from set theory. The *cumulative hierarchy (over atoms $\mathbb{A}$)* is indexed by ordinal numbers, and defined as follows: on level 0 we find the atoms, and on level indexed by an ordinal $\alpha > 0$ we find the atoms and every set which has all elements taken from levels $< \alpha$. For example, on level 1 we find every subset of the atoms. Automorphisms $\pi : \mathbb{A} \to \mathbb{A}$ of the atoms act on sets in the cumulative hierarchy in the expected way.

In the cumulative hierarchy one can encode data structures such as tuples, words, relations, functions etc., using standard set-theoretic machinery. Automorphisms π then act on tuples as expected: $\pi((x_1, \ldots, x_n)) = (\pi(x_1), \ldots, \pi(x_n))$.

For a tuple of atoms $\bar{a} \in \mathbb{A}^*$, an *$\bar{a}$-automorphism* is an atom automorphism π such that $\pi(\bar{a}) = \bar{a}$. A tuple $\bar{a}$ is said to *support* an element x of the cumulative hierarchy if $\pi(x) = x$ for every $\bar{a}$-automorpism π. We say that an element x of the cumulative hierarchy is *finitely supported* if it is supported by some tuple of atoms in $\mathbb{A}^*$. An element is called *equivariant* if it is supported by the empty tuple. An equivariant set may contain non-equivariant elements; e.g., the set of all atoms is equivariant, but its elements are not. A *set with atoms* is defined to be a set in the cumulative hierarchy which is finitely supported and with each of its members (and their members, and so on) finitely supported.

Sets with atoms provide a relaxed notion of finiteness, called orbit-finiteness. We say that two sets with atoms x and y are in the same $\bar{a}$-orbit if there is some $\bar{a}$-automorphism π such that $y = \pi(x)$. Being in the same $\bar{a}$-orbit is clearly an equivalence relation; its equivalence classes are called *$\bar{a}$-orbits*. A set with atoms x is called *orbit-finite* if there is some atom tuple $\bar{a}$ such that x is a finite union of $\bar{a}$-orbits.

Example 1.

- Over equality atoms, the set $\mathbb{A}$ of atoms is orbit-finite, also for every $k \in \{1, 2, \ldots\}$ the sets $\mathbb{A}^{(k)}$ and $\binom{\mathbb{A}}{k}$ from (1) are orbit-finite, with a single equivariant orbit each. The set $\mathbb{A}^k$ is also orbit-finite, with the number of equivariant orbits equal to the k-th Bell number.
- Over ordered atoms, the sets $\mathbb{A}$ and $\binom{\mathbb{A}}{k}$ have single equivariant orbit each. The set $\mathbb{A}^{(k)}$ is also orbit-finite, with $k!$ equivariant orbits; one of these orbits is the increasing k-tuples of atoms. Each of these orbits is in an equivariant bijection with $\binom{\mathbb{A}}{k}$.

- Over the ordered atoms, for every atom $a \in \mathbb{A}$ the set $\mathbb{A} - \{a\}$ has two a-orbits: the open intervals $(-\infty; a)$ and $(a; \infty)$.
- The set $\mathbb{A}^*$ is not orbit-finite, neither for equality nor for ordered atoms, as words of different length are necessarily in distinct orbits. □

An atom structure $\mathbb{A}$ is called *oligomorphic* if for every k, the set $\mathbb{A}^k$ is orbit-finite. For oligomorphic atoms, orbit-finite sets behave well, e.g. they are closed under products and finitely supported subsets [3, Lem. 3.24]. In this paper we only consider oligomorphic atoms.

Orbit-finite sets can be represented in a finite way so that they can be used as inputs for algorithms. Two examples of such representations are set builder expressions [3, Sec. 4.1] or the G-set representation [4, Sec. 8]. The reader does not need to know these representations in detail; the important thing is that they support basic operations such as products, Boolean operations, or inclusion and membership checks.

III. Weighted orbit-finite automata

We now introduce the main model for this paper, an orbit-finite generalization of weighted automata.

Definition III.1 (Weighted orbit-finite automaton) *Fix an oligomorphic atom structure $\mathbb{A}$ and a field $\mathbb{F}$. A* weighted orbit-finite automaton *consists of orbit-finite sets Q and Σ, called the* states and alphabet*, and finitely supported functions*

$$\underbrace{I : Q \to \mathbb{F}}_{\textit{initial}} \quad \underbrace{\delta : Q \times \Sigma \times Q \to \mathbb{F}}_{\textit{transitions}} \quad \underbrace{F : Q \to \mathbb{F}}_{\textit{final}}.$$

Furthermore, we require the non-guessing condition:

(*) *there are finitely many states with nonzero initial weight, and also for every state q and input letter a, there are finitely many states p such that the transition (q, a, p) has nonzero weight.*

Weights of runs and words is defined in the same way as for the classical model of weighted automata with finitely many states. The non-guessing condition ensures that each word has finitely many runs; otherwise there could be difficulties in summing up the weights of infinitely many runs.

Example 2. Consider any oligomorphic atoms $\mathbb{A}$ and the field of rational numbers. We define a weighted orbit-finite automaton which maps a word $w \in \mathbb{A}^*$ to the number of distinct atoms that appear in w. The states are

$$\underbrace{\{\bot\}}_{\substack{\text{initial weight 1}\\ \text{final weight 0}}} \quad + \quad \underbrace{\mathbb{A}}_{\substack{\text{initial weight 0}\\ \text{final weight 1}}}$$

for some $\bot \notin \mathbb{A}$. The transition weight is 1 for all triples

$$(\bot, a, \bot)\ (\bot, a, a)\ (a, b, a) \qquad \text{for } a \neq b \in \mathbb{A},$$

and 0 for all other triples. For every input word, all runs have weight 0, except for the following runs, which have weight 1: start in $\bot$, stay there until the last occurrence of some atom a, and then stay in state a until the end of the word. Since the number of runs with weight 1 is the number of distinct atoms a that appear in the word, the output of the automaton is the number of distinct atoms.

This automaton is equivariant, in the sense that its state space and all three weight functions are equivariant. □

Example 3. The automaton in the previous example is a special case of a more general construction: counting accepting runs of a nondeterministic automaton. Define a *nondeterministic orbit-finite automaton* like a nondeterministic finite automaton, except that all components (alphabet, states, transitions, initial and finite sets) are required to be orbit-finite sets, see [3, Def. 5.7]. To such an automaton one can associate a weighted orbit-finite automaton (over the field of rational numbers) as follows: the alphabet and states are the same, and the remaining components are defined by replacing "yes" with weight 1 and "no" with weight 0. This weighted automaton maps a word to the number of accepting runs of the nondeterministic automaton. The construction makes sense only if the original nondeterministic automaton is non-guessing in the sense that there are finitely many inputs states, and for every pair (state, letter) there are finitely many outgoing transitions. □

As mentioned in Section II, orbit-finite sets can be represented in a finite way. Therefore, weighted orbit-finite automata can also be represented in a finite way (assuming that field elements can be represented in a finite way). This is because (a) a finitely supported relation on an orbit-finite set is also orbit-finite; and (b) a finitely supported function from an orbit-finite set to any set is itself an orbit-finite set.

Example 4. A special case of weighted orbit-finite automata, where the finite representation is easier to see, is a weighted k-register automaton. This is a weighted orbit-finite automaton where the input alphabet is finitely many disjoint copies of the atoms, the states are finitely many copies of $(\mathbb{A} + \{\bot\})^k$, and all weight functions are equivariant. For equality and ordered atoms, the weight functions can be finitely represented using quantifier-free formulas, see [3, p. 6]. □

We now state the main result of this paper, which is an algorithm for equivalence problem of weighted orbit-finite automata, assuming that the atom structure is either the equality atoms or the ordered atoms. We do not know if the problem is decidable for other atom structures.

Theorem III.2 *Assume that the atoms are the equality atoms $(\mathbb{N}, =)$ or the ordered atoms $(\mathbb{Q}, <)$. The equivalence problem for equivariant*[1] *weighted orbit-finite automata can be solved in deterministic time*

$$2^{\mathrm{poly}(k)} \cdot n^{O(k)}$$

where

[1]This theorem would also work for finitely supported automata, but the notation for the complexity and orbit counts would be more involved.

- n *is the orbit count, i.e. the number of equivariant orbits in the disjoint union of the two state sets;*
- k *is the atom dimension of the state spaces of the automata, i.e. the smallest k such that every state in both automata is supported by at most k atoms.*

In particular, the equivalence problem is in EXPTIME*, and polynomial time when the atom dimension k is fixed.*

A lower bound for the problem is PSPACE, which is the complexity of language equivalence of deterministic register automata. We do not know the exact complexity; it is worth pointing out that for weighted finite automata there is an equivalence algorithm in randomized polylogarithmic parallel time [11, Sec. 3.2] that uses the Isolating Lemma.

We now present a proof strategy for the theorem, which will be carried out in the next sections. The first observation is that the equivalence problem reduces to the zeroness problem, which asks whether a single weighted orbit-finite automaton outputs zero for every word. The reduction is as follows: given two equivariant weighted orbit-finite automata $\mathcal{A}_1$ and $\mathcal{A}_2$, we create a new weighted orbit-finite automaton $\mathcal{A}_1 - \mathcal{A}_2$, which is obtained by taking the disjoint union of $\mathcal{A}_1$ and $\mathcal{A}_2$, and flipping the sign of the final weights in $\mathcal{A}_2$. The new automaton maps all words to zero if and only if the original two automata were equivalent. Also, in the reduction, the atom dimension does not change, neither does the orbit count.

It remains to give an algorithm for zeroness of a single equivariant weighted orbit-finite automaton, with states Q. Our proof follows the same lines as Schützenberger's algorithm. Write $\mathsf{Lin}\, Q$ for the set of finite linear combinations (over the field $\mathbb{F}$) of states, seen as a vector space with basis Q. For an input word w, define its *configuration* to be the vector in $\mathsf{Lin}\, Q$ which maps a state q to the sum of pre-weights of all runs over w that end in state q; the pre-weight of a run is defined by multiplying the initial weight of the first state and the weights of all transitions, without taking into account the final weights. Thanks to condition (*) in Definition III.1, each configuration is indeed a finite linear combination, since there are finitely many runs with nonzero pre-weight. Consider the chain

$$V_0 \subseteq V_1 \subseteq V_2 \subseteq \cdots \subseteq \mathsf{Lin}\, Q$$

where V_i is the subspace of $\mathsf{Lin}\, Q$ that is spanned by the configurations of words of length at most i. (We do not intend to compute the subspaces in this chain, whatever that would mean; the chain is only used in the analysis of the algorithm.) Because the automaton is equivariant, one can easily see that each subspace V_n is also equivariant.

In the finite-dimensional case studied by Schützenberger, where Q was a finite set, we could conclude that the chain must stabilize in a number of steps that is bounded by the finite dimension of the vector space $\mathsf{Lin}\, Q$. In the orbit-finite case, the vector space has infinite dimension, and thus it is not clear why the chain should stabilize in finitely many steps. Our main technical contribution is a proof that the chain does indeed stabilize, and furthermore the time to stabilize is consistent with the bounds in the statement of Theorem III.2. This stabilization property is the subject of the next section, and in the section after that we use the property to conclude the proof of Theorem III.2.

IV. Finite length property

A standard result in linear algebra says that a vector space has finite dimension if and only if it has finite length, where the length is defined to be the longest chain of its subspaces. Because of this easy correspondence the notion of length is seldom explicitly applied to vector spaces, and it becomes more important only in more general structures such as modules over a ring. However, the situation becomes more interesting for chains of *equivariant* subspaces. We define:

Definition IV.1 (Length) *The* length *of an equivariant vector space V, denoted* $\mathsf{length}(V)$*, is the maximal length n of a chain of proper inclusions on equivariant subspaces of V:*

$$V_1 \subsetneq V_2 \subsetneq \cdots \subsetneq V_n \subsetneq V.$$

If a maximal length does not exist, we say that V has infinite length. An atom structure $\mathbb{A}$ has the finite length property *if for every number k, the vector space $\mathsf{Lin}\, \mathbb{A}^k$ has finite length.*

The finite length property easily implies oligomorphicity, but we do not know if the converse implication holds. The purpose of this section is to prove that both the equality atoms $(\mathbb{N}, =)$ and the ordered atoms $(\mathbb{Q}, <)$ have the finite length property (Lemmas IV.9 and IV.8). Furthermore, the length of $\mathsf{Lin}\, \mathbb{A}^k$ grows exponentially (and not worse) with k. The results of this section apply to an arbitrary field $\mathbb{F}$.

Definition IV.1 speaks of equivariant subspaces. At the end of this section, we show that for the equality and ordered atoms, allowing a fixed support would make no difference.

Example 5. Consider the equality atoms. The vector space $\mathsf{Lin}\, \mathbb{A}$ has length 2, because it has exactly 3 equivariant subspaces which form a chain of two proper inclusions:

$$\{0\} \quad \subsetneq \quad \underbrace{\mathsf{Span}\{a - b : a \neq b \in \mathbb{A}\}}_{\text{call this } V} \quad \subsetneq \quad \mathsf{Lin}\, \mathbb{A}.$$

Equivalently, V is the vector space of all vectors where all coefficients sum up to 0. Let us prove that there are no other equivariant subspaces. Suppose that W is an equivariant subspace which contains some nonzero vector

$$w = \lambda_1 a_1 + \cdots + \lambda_n a_n \quad \text{where } \lambda_i \in \mathbb{F} \setminus \{0\}. \qquad (2)$$

Since W is equivariant, it also contains the vector obtained from w by replacing a_n with some fresh atom b_n. By taking the difference of these vectors and dividing by λ_n, we see that W contains $a_n - b_n$. By equivariance, W contains all vectors of the form $a - b$ for distinct atoms a, b, and thus $V \subseteq W$.

We now show that W is either V or the entire space $\mathsf{Lin}\, \mathbb{A}$. Indeed, suppose that W contains some w as in (2) that is not in V. If $n > 1$ then we can subtract from w the vector $\lambda_1 \cdot (a_1 - a_2) \in V$, which results in another vector that is in W

but not in V, with a smaller n. By repeating this process, we see that W contains a vector of the form $\lambda_1 a_1$ with $\lambda_1 \neq 0$, and hence, by equivariance, it is the entire space. □

Definition IV.1 differs from the classical definition of length in that we only consider equivariant subspaces. Some classical properties of length easily transfer to our case. One may even say that our definition is a special case of the classical one: an equivariant vector space can be seen as a module over the (non-commutative) group ring $\mathbb{F}[G]$, where $\mathbb{F}$ is the underlying field and G is the automorphism group of $\mathbb{A}$. To keep the presentation elementary we do not pursue this correspondence, but we remark that all properties of length which are valid for modules over (non-commutative) rings, remain true for our definition. For example, the following lemma has the same proof (see Appendix A) as for the classical notion of length of a module, (see e.g. [9, Prop. 4.12]), and works for arbitrary oligomorphic atoms:

Lemma IV.2 *For any equivariant spaces $V \subseteq W$, and equivariant sets P, Q with their disjoint union $P + Q$:*

(i) $\mathsf{length}(W) = \mathsf{length}(V) + \mathsf{length}(W/V)$*;*
(ii) $\mathsf{length}(\mathsf{Lin}(P + Q)) = \mathsf{length}(\mathsf{Lin}\, P) + \mathsf{length}(\mathsf{Lin}\, Q)$*;*
(iii) *if there is an equivariant surjective function from P to Q then* $\mathsf{length}(\mathsf{Lin}\, Q) \leq \mathsf{length}(\mathsf{Lin}\, P)$.

From this we infer that in Definition IV.1, we could have equivalently talked about arbitrary equivariant orbit-finite set, instead of just sets of the form $\mathbb{A}^k$:

Corollary IV.3 *For atoms with the finite length property,* $\mathsf{Lin}\, Q$ *has finite length for every equivariant orbit-finite Q.*

Proof
An orbit-finite set is a finite disjoint union of single-orbit sets, and for oligomorphic atoms every single-orbit set Q is an image of a surjective function from $\mathbb{A}^k$ [5], for k the atom dimension of Q. Hence, we can use the closure properties from Lemma IV.2. □

The following lemma is the key step in proving the finite length property for the equality and ordered atoms:

Lemma IV.4 *Consider the ordered atoms $\mathbb{A} = (\mathbb{Q}, <)$. For every k, the length of* $\mathsf{Lin}\binom{\mathbb{A}}{k}$ *is finite, and satisfies*

$$\mathsf{length}\left(\mathsf{Lin}\binom{\mathbb{A}}{k}\right) \quad \leq \quad 1 + k \cdot \mathsf{length}\left(\mathsf{Lin}\binom{\mathbb{A}}{k-1}\right).$$

Proof
For any set α of $2k$ atoms:

$$\underbrace{a_1 < b_1 < \cdots < a_k < b_k}_{\alpha} \tag{3}$$

and for any $I \subseteq k = \{1, \ldots, k\}$, define $\alpha \rtimes I \in \binom{\mathbb{A}}{k}$ by:

$$\alpha \rtimes I = \{a_i \mid i \notin I\} \cup \{b_i \mid i \in I\}.$$

In words, $\alpha \rtimes I$ picks either a_i or b_i from α according to I. Define the *cog* (on α), $\nu^\alpha \in \mathsf{Lin}\binom{\mathbb{A}}{k}$, to be the vector:

$$\nu^\alpha = \sum_{I \subseteq k} (-1)^{|I|} (\alpha \rtimes I).$$

Notice that all cogs form a single orbit in $\mathsf{Lin}\binom{\mathbb{A}}{k}$.

Claim IV.5 *Every nontrivial equivariant subspace $V \subseteq$* $\mathsf{Lin}\binom{\mathbb{A}}{k}$ *contains a cog.*

Proof
Pick any nonzero $v \in V$ and pick $\alpha = \{a_1, \ldots, a_k\} \in \binom{\mathbb{A}}{k}$ so that $v(\alpha) \neq 0$. Choose fresh atoms $b_1, \ldots, b_k$ to form a set as in (3). For every $i = 1, \ldots, k$, choose an atom automorphism π_i such that:

- $\pi_i(a_i) = b_i$, and
- $\pi_i(a_j) = a_j$ and $\pi_i(b_j) = b_j$ for $j \neq i$.

Then put $v_0 = v$ and define $v_1, \ldots, v_k \in V$ by induction:

$$v_i = v_{i-1} - \pi_i(v_{i-1})$$

($\pi(v_{i-1}) \in V$ since V is equivariant). It is easy to prove by induction that $v_i(\alpha) = v(\alpha)$ for all i, and so in particular all v_i are nonzero. Also by straightforward induction, each v_i has the following properties:

- $v_i(\beta)$ is nonzero only if for each $1 \leq j \leq i$, β contains either a_j or b_j but not both,
- if β' arises from such β by replacing a_j with b_j (or *vice versa*) for exactly one $j \leq i$, while keeping the other components unchanged, then $v_i(\beta) + v_i(\beta') = 0$.

For $i = k$ this implies that v_k, divided by the scalar $v(\alpha)$, is a cog. □

Claim IV.5 implies that $\mathsf{Lin}\binom{\mathbb{A}}{k}$ has a unique least nontrivial equivariant subspace: the one spanned by all cogs. We shall now give an explicit description of that subspace.

A vector $v \in \mathsf{Lin}\binom{\mathbb{A}}{k}$ is called *balanced* if for every set $S \in \binom{\mathbb{A}}{k-1}$, and for every S-orbit $I \subseteq \mathbb{A}$ such that $I \cap S = \emptyset$:

$$\textstyle\sum_{a \in I} v(S \cup \{a\}) = 0. \tag{4}$$

Note that if $S = \{a_1, \ldots, a_{k-1}\}$ where $a_1 < \cdots < a_{k-1}$, then S-orbits disjoint from S are exactly the k open intervals:

$$(-\infty, a_1),\ (a_1, a_2), \ldots, (a_{k-2}, a_{k-1}),\ (a_{k-1}, +\infty). \tag{5}$$

It is easy to check that balanced vectors form an equivariant subspace of $\mathsf{Lin}\binom{\mathbb{A}}{k}$. We denote this space by B. An immediate corollary of Claim IV.5 is that every cog is balanced.

We will show that B is the subspace generated by all cogs. For this, it is enough to show that every balanced vector is a linear combination of cogs. To this end, we will prove that every balanced vector supported by some fixed $T = \{a_1, \ldots, a_n\}$ (that is, every vector in $B \cap \mathsf{Lin}\binom{T}{k}$) is a linear combination of cogs supported by T (or, in short, *cogs over T*). We shall do this by calculating the (finite!) dimension of the space $B \cap \mathsf{Lin}\binom{T}{k}$. But first, let us count the cogs:

Claim IV.6 *There are at least $\binom{n-k}{k}$ linearly independent cogs over T.*

Proof
Call a set α as in (3) *narrow* if no element in T lies in between a_i and b_i for any i. For a narrow α, the cog ν^α will also be called narrow.

There are exactly $\binom{n-k}{k}$ narrow sets in T: choosing a narrow set from the n-element set T amounts to choosing k elements from an $(n-k)$-element set, and then viewing the chosen elements as neighbouring pairs (a_i, b_i) in the narrow set.

All narrow cogs over T are linearly independent. To see why, consider the componentwise partial order $\sqsubseteq$ on $\binom{T}{k}$:

$$(a_1 < \cdots < a_k) \sqsubseteq (a'_1 < \cdots < a'_k) \quad \text{if} \quad a_i \leq a'_i \quad \text{for all } i.$$

Note that on narrow sets, the mapping:

$$\underbrace{a_1 < b_1 < \cdots < a_k < b_k}_{\alpha} \quad \mapsto \quad \underbrace{a_1 < \cdots < a_k}_{\alpha \rtimes \emptyset}$$

is injective, so $\sqsubseteq$ lifts to a partial order on the set of narrow sets, and hence on the set of narrow cogs.

Assume that narrow cogs are linearly dependent, and so there is some linear combination:

$$\sum_\alpha p_\alpha \nu^\alpha = 0 \tag{6}$$

where the sum ranges over all narrow sets α in T, and $p_\alpha \in \mathbb{F}$. Note that this equation lives in the vector space $\mathsf{Lin}\binom{T}{k}$.

Let α be a minimal (with respect to $\sqsubseteq$) narrow set such that $p_\alpha \neq 0$. Notice that, for every α, ν^α is the greatest (with respect to $\sqsubseteq$) narrow cog in which $\alpha \rtimes \emptyset$ appears with a nonzero coefficient. Since every narrow cog smaller than ν^α has coefficient zero on the left-hand side of (6), then $\alpha \rtimes \emptyset$ has a nonzero coefficient in the *sum* of the left-hand side of (6), which contradicts the equation in (6). $\square$

The main technical result in the proof of the present lemma is the following claim, which is proved in Appendix B.

Claim IV.7 $B \cap \mathsf{Lin}\binom{T}{k}$ *is of dimension at most* $\binom{n-k}{k}$.

Claims IV.6 and IV.7 together imply that B is the space generated by all cogs, and therefore – by Claim IV.5 – that it is the smallest nontrivial equivariant subspace of $\mathsf{Lin}\binom{\mathbb{A}}{k}$.

To finish the proof of Lemma IV.4, for every $i \in \{1, \ldots, k\}$ consider the equivariant linear map:

$$g_i : \mathsf{Lin}\binom{\mathbb{A}}{k} \to \mathsf{Lin}\binom{\mathbb{A}}{k-1}$$

defined by:

$$g_i(v)(S) = \textstyle\sum_{a \in I} v(S \cup \{a\})$$

where I is the i'th orbit on the list (5). Tupling these functions for all i we obtain an equivariant linear map:

$$g : \mathsf{Lin}\binom{\mathbb{A}}{k} \to \left(\mathsf{Lin}\binom{\mathbb{A}}{k-1}\right)^k.$$

By definition, the kernel of g is B, so there is a subspace embedding:

$$\left(\mathsf{Lin}\binom{\mathbb{A}}{k}\right)/B \subseteq \left(\mathsf{Lin}\binom{\mathbb{A}}{k-1}\right)^k.$$

Note that B does not have any nontrivial equivariant subspaces, so $\mathsf{length}(B) = 1$. By Lemma IV.2(i) applied to $V = B$ and $W = \mathsf{Lin}\binom{\mathbb{A}}{k}$ we thus obtain:

$$\mathsf{length}\left(\mathsf{Lin}\binom{\mathbb{A}}{k}\right) = 1 + \mathsf{length}\left(\left(\mathsf{Lin}\binom{\mathbb{A}}{k}\right)/B\right) \leq$$

$$\leq 1 + \mathsf{length}\left(\left(\mathsf{Lin}\binom{\mathbb{A}}{k-1}\right)^k\right) = 1 + k \cdot \mathsf{length}\left(\mathsf{Lin}\binom{\mathbb{A}}{k-1}\right)$$

and conclude the proof of Lemma IV.4. $\square$

The following is now easy:

Lemma IV.8 *The ordered atoms $\mathbb{A} = (\mathbb{Q}, <)$ have the finite length property. For an equivariant orbit-finite set Q, the space $\mathsf{Lin}\,Q$ has length at most*

$$\textit{(orbit count of } Q\textit{)} \cdot (1 + \textit{atom dimension of } Q)!$$

Proof
If Q has only one orbit, then it is an image of an equivariant surjective function from $\binom{\mathbb{A}}{k}$, where k is the atom dimension of Q. By induction on k, using Lemma IV.4, we obtain

$$\mathsf{length}\left(\mathsf{Lin}\binom{\mathbb{A}}{k}\right) \leq (1+k)! \tag{7}$$

The lemma then follows by Lemma IV.2(iii).

Finally, the case of multi-orbit Q follows from Lemma IV.2(ii). $\square$

From this it is easy to deduce the finite length property for equality atoms.

Lemma IV.9 *The equality atoms $\mathbb{A} = (\mathbb{N}, =)$ have the finite length property. For an equivariant orbit-finite set Q, the space $\mathsf{Lin}\,Q$ has length at most*

$$\textit{(orbit count of } Q\textit{)} \cdot k! \cdot (1+k)!$$

where k is the atom dimension of Q.

Proof
The set $\mathbb{A}^{(k)}$, a single-orbit set over equality atoms, can be seen as an equivariant set over ordered atoms as well, with $k!$ disjoint orbits. Each of these orbits is equivariantly (over ordered atoms) isomorphic to $\binom{\mathbb{A}}{k}$. Lemma IV.8 then implies that $\mathsf{Lin}\left(\mathbb{A}^{(k)}\right)$ has length at most $k! \cdot (1+k)!$.

Every single-orbit equivariant set Q is an image of an equivariant surjective function from $\mathbb{A}^{(k)}$, where k is the atom dimension of Q; the lemma follows by Lemma IV.2(iii). The case of multi-orbit Q follows from Lemma IV.2(ii). $\square$

So far, we have bounded the lengths of chains of equivariant subspaces. Lemma IV.10 below, proved in Appendix C, extends these bounds to finitely supported chains. Here it is important that all subspaces the chain are required to have the same support.

Lemma IV.10 *Consider the equality or ordered atoms. Let $\bar{a}$ be a tuple of atoms. For every $\bar{a}$-supported orbit-finite set Q, there is a finite upper bound on the length of chains of $\bar{a}$-supported subspaces of* $\mathsf{Lin}\, Q$.

V. The equivalence algorithm

In this section, we use the finite length properties from Section IV to complete the proof of Theorem III.2. We assume that the atoms $\mathbb{A}$ are the equality or ordered atoms.

Lemma V.1 *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be equivariant weighted orbit-finite automata, and let n and k be as in Theorem III.2. If the recognized weighted languages are different, then this difference is witnessed by some input word of length at most*

$$2^{\mathrm{poly}(k)} \cdot n.$$

Proof
Let Q be the state space of the difference automaton $\mathcal{A}_1 - \mathcal{A}_2$ as described in the reduction from equivalence to zeroness. For an input word $w \in \Sigma^*$, let $[w] \in \mathsf{Lin}\, Q$ be its corresponding configuration in the difference automaton, and let $V_i \subseteq \mathsf{Lin}\, Q$ be the subspace that is spanned by configurations of input words of length at most i. By Lemmas IV.9 and IV.8, we know that the chain $\{V_i\}_i$ must stabilize in a number of steps that is bounded as in the statement of the lemma. This means that for every input word, its configuration is a linear combination of configurations of short words; in particular the difference automaton can produce a nonzero output if and only if it can produce a nonzero output on a short input word. □

At this point, we could solve the equivalence problem by guessing a short differentiating word, leading to a nondeterministic algorithm for non-equivalence, with running time as in the bound from Lemma V.1. We can, however, improve this to get a deterministic algorithm, by using a reduction to the equivalence problem for finite weighted automata. Short input words necessarily use few atoms, and therefore the equivalence problem boils down to testing equivalence for input words that have few atoms. The latter problem is solved in the following lemma.

Lemma V.2 *Consider the following problem.*

- **Input.** *Two equivariant weighted orbit-finite automata $\mathcal{A}_1$ and $\mathcal{A}_2$ and a number $\ell \in \{1, 2, \ldots\}$;*
- **Question.** *Are the two automata equivalent on all input words supported by at most ℓ atoms?*

This problem can be solved in time polynomial in $n \cdot \ell^k$, where the parameters n and k are defined as in Theorem III.2.

Proof
Choose a tuple $\bar{a}$ of ℓ atoms. Since the automata are equivariant, they are equivalent on inputs with at most ℓ atoms if and only if they are equivalent on inputs words supported by $\bar{a}$.

By equivariance and condition (*) from Definition III.1, if a state q and an input letter σ are both supported by $\bar{a}$, then the same is true for every state p such that (q, σ, p) is a transition with nonzero weight. Therefore, when restricted to input words that are supported by $\bar{a}$, both automata only use states that are supported by $\bar{a}$. These observations motivate the following definition: for $i \in \{1, 2\}$, let $\mathcal{A}_{i,\bar{a}}$ the weighted automaton that is obtained from $\mathcal{A}_i$ by restricting the states and alphabet to elements supported by $\bar{a}$, and restricting the transitions and weight functions to the new alphabet and states. These automata are finite, with size bounded by $n \cdot \ell^k$. The lemma follows by applying the polynomial time algorithm for equivalence of finite weighted automata. □

The above two lemmas complete the proof of Theorem III.2. The algorithm for zeroness is simply the algorithm from Lemma V.2, with ℓ being the bound from Lemma V.1.

VI. Vector spaces with atoms

In this section, we study in more depth vector spaces that are spanned by orbit-finite sets. Apart from their independent interest, these results will be used in Section VII to minimize weighted automata, and in Section VIII to decide equivalence for unambiguous automata.

So far we have discussed vector spaces of the form $\mathsf{Lin}\, Q$, where Q is an orbit-finite set. One could call these spaces *orbit-finite-dimensional*, since they have an orbit-finite basis. Although they are good enough to treat weighted orbit-finite automata, such vector spaces are not very robust. The reason is that extracting a basis uses choice, see [2, Thm. 1], and the principle of choice fails for sets with atoms.

Example 6. Recall the vector space $\mathsf{Lin}\, \mathbb{A}$ that was discussed in Example 5, and the subspace V that was spanned by

$$X = \{a - b : a \neq b \in \mathbb{A}\}.$$

The set X is not a basis, since the vectors $a - b$ and $b - a$ (or $a - b$, $b - c$ and $a - c$) are linearly dependent. However, X is a single equivariant orbit, so it does not have any nonempty proper subset that is equivariant. Therefore, no equivariant subset of X is a basis. In fact, V does not have any equivariant basis, even if we allow bases that are not contained in X. It does have, however, a finitely supported basis contained in X: for any fixed atom $a_0 \in \mathbb{A}$, the set

$$\{a_0 - b : b \in \mathbb{A} - \{a_0\}\}$$

is a basis. □

In the previous example, there was no equivariant basis, but there was a finitely supported one. In the next example there is no finitely supported basis at all.

Example 7. Consider the equality atoms and the space $\mathsf{Lin}(\mathbb{A}^{(2)})$. A vector in this space can be visualized as a weighted directed finite graph, where vertices are atoms and the weight of an edge (a, b) is the corresponding coefficient in the vector. Consider the subspace of $\mathsf{Lin}(\mathbb{A}^{(2)})$ spanned by

$$X = \{(a, b) - (b, a) : a \neq b \in \mathbb{A}\}.$$

This subspace consists of graphs where for every pair of vertices, the connecting edges in both directions have opposite weights. We claim that there is no finitely supported subset of X that is a basis of this space. Indeed, suppose that $Y \subseteq X$ is a finitely supported basis. It is easy to see that for every two distinct atoms a, b, the set Y must contain one of the vectors

$$(a,b)-(b,a) \qquad \text{or} \qquad (b,a)-(a,b).$$

If the atoms a, b are fresh (i.e. they do not belong to the least support of Y), by swapping these two atoms we map one of the vectors to the other, hence *both* vectors must belong to Y. However, the two vectors are linearly dependent and so Y is not a basis. Using a similar argument one can show that the subspace spanned by X does not have any finitely supported basis, even if we allow bases that are not contained in X. □

The above example shows that vector spaces with an orbit-finite basis are not closed under finitely supported subspaces. The same issue appears with finitely supported quotients (images of surjective linear maps). These issues will become a problem in the next section, where we minimize weighted orbit-finite automata, since minimization will require taking subspaces and quotients. To deal with these issues, we propose an obvious generalization: vector spaces that are spanned by an orbit-finite set (which need not be independent).

Definition VI.1 (Orbit-finitely spanned vector space) *Fix a field* $\mathbb{F}$ *and an atom structure. A* vector space with atoms *is a vector space such that the underlying set* V *is a set with atoms, and the operations*

$$\underbrace{+ : V \times V \to V}_{\textit{binary addition}} \qquad \underbrace{\cdot : \mathbb{F} \times V \to V}_{\textit{scalar multiplication}}$$

are finitely supported. A vector space with atoms is called orbit-finitely spanned *if there is an orbit-finite subset of the vector space which spans it.*

As we have seen in Example 7, some orbit-finitely spanned vector spaces do not have any orbit-finite basis. They do, however, have finite length:

Lemma VI.2 *For atoms with the finite length property, equivariant orbit-finitely spanned vector spaces have finite length.*

Proof
Let V be spanned by an equivariant orbit-finite set Q, and let

$$f : \ \mathsf{Lin}\, Q \to V$$

be the unique linear map which extends the inclusion of Q in V. This is a surjective finitely supported linear map, so (by Lemma IV.2(iii)) the length of V is not greater than the length of $\mathsf{Lin}\, Q$, which is finite by the finite length property. □

Theorem VI.3 (Closure properties) *Assume that the atoms have the finite length property. Equivariant orbit-finitely spanned vector spaces are closed under equivariant quotients, equivariant subspaces, direct sums and tensor products.*

Proof
Quotients and direct sums are immediate. For the tensor product $U \otimes V$, suppose U and V are spanned by Q and P. Since $\mathsf{Lin}\, Q \otimes \mathsf{Lin}\, P$ is isomorphic to $\mathsf{Lin}(Q \times P)$, it follows that $U \otimes V$ is spanned by $Q \times P$, which is orbit-finite.

We are left with the subspaces. Suppose that V is equivariant and orbit-finitely spanned, and let U be an equivariant subspace. For $n \in \{1, 2, \ldots\}$ define U_n to be the subspace of U that is spanned by vectors from U that are supported by at most n atoms. This is a chain

$$U_1 \subseteq U_2 \subseteq \cdots$$

of equivariant vector spaces, and therefore by the finite length property it stabilizes after finitely many steps. It remains to show that every U_n is orbit-finitely spanned. For every tuple $\bar{b}$ of n atoms, the vectors in V that are supported by $\bar{b}$ form a finite-dimensional vector space, and so they have a finite basis. By closing this basis under atom automorphisms, we get an orbit-finite spanning set for U_n. □

Thanks to Lemma IV.10, in the equality and ordered atoms, Lemma VI.2 and Theorem VI.3 hold also for orbit-finitely spanned vector spaces that are not necessarily equivariant.

In the remainder of this section, we discuss another closure property of orbit-finitely spanned vector spaces: if V is an equivariant orbit-finitely spanned vector space, then the same is true for its *finitely supported dual*, which is the vector space of finitely supported linear maps from V to $\mathbb{F}$. We will prove closure under finitely supported duals for the equality and ordered atoms, and we do not know if it holds for other oligomorphic atoms.

Before considering finitely supported duals, we discuss a slightly simpler vector space, namely the finitely supported functions from some orbit-finite set to the underlying field. Let Q be an orbit-finite set. Define

$$Q \xrightarrow{\text{fs}} \mathbb{F}$$

to be the set of finitely supported functions from Q to the field. This set can be viewed as a vector space, with coordinate-wise addition and scalar multiplication. We have

$$\mathsf{Lin}\, Q \ \subseteq \ Q \xrightarrow{\text{fs}} \mathbb{F},$$

and the inclusion is strict when Q is infinite, as witnessed by the function that returns 1 on all arguments.

Example 8. For the equality atoms, the space $\mathbb{A} \xrightarrow{\text{fs}} \mathbb{F}$ is spanned by the functions

$$\{f_{\mathbb{A}}\} \ \cup \ \{f_a : a \in \mathbb{A}\} \tag{8}$$

where $f_{\mathbb{A}}$ maps all atoms to 1, while f_a maps a to 1 and the remaining atoms to 0. Indeed, every function $f : \mathbb{A} \to \mathbb{F}$ supported by an atom tuple $\bar{a}$ can be presented as:

$$f \ = \ \underbrace{f(c)}_{\substack{c \text{ is some fresh} \\ \text{atom not in } \bar{a}}} \cdot f_{\mathbb{A}} \ + \ \sum_{a \in \bar{a}} (f(a) - f(c)) \cdot f_a.$$

The functions from (8) are linearly independent, and therefore they form an orbit-finite basis of $\mathbb{A} \xrightarrow{\text{fs}} \mathbb{F}$.

A similar result holds for the ordered atoms, except that there the spanning set of functions is

$$\{f_{\mathbb{A}}\} \ \cup\ \{f_a : a \in \mathbb{A}\} \ \cup\ \{f_{>a} : a \in \mathbb{A}\}$$

where $f_{\mathbb{A}}$ and f_a are as before and $f_{>a}$ maps b to 1 precisely if $b > a$. $\Box$

The above example shows that $\mathbb{A} \xrightarrow{\text{fs}} \mathbb{F}$ is orbit-finitely spanned. This is true for every orbit-finite set, not just $\mathbb{A}$:

Theorem VI.4 *Assume the equality or the ordered atoms. If Q is an orbit-finite set, then $Q \xrightarrow{\text{fs}} \mathbb{F}$ is orbit-finitely spanned.*

Proof
To simplify notation, we assume that Q is equivariant; this implies the general case anyway, because every orbit-finite set is contained in some equivariant orbit-finite set. Furthermore, it is safe to assume that Q is a single-orbit set.

We need to exhibit an orbit-finite set Φ of finitely supported functions, so that every finitely supported function $f : Q \to \mathbb{F}$ is a linear combination of functions from Φ. It is enough to show this for the case where f is the characteristic function of a finitely supported set $R \subseteq Q$, since such characteristic functions are easily seen to span the space of all finitely supported functions. So consider such a set R, supported by some tuple $\bar{a}$ of atoms. R is then a disjoint union of $\bar{a}$-orbits, and it is enough to consider the case where R is a single $\bar{a}$-orbit.

From here, the arguments for the equality and the ordered atoms begin to differ.

For the case of **ordered atoms**, we know that Q is in equivariant bijection with $\binom{\mathbb{A}}{k}$ for some number k, so it is enough to deal with $Q = \binom{\mathbb{A}}{k}$. Let R be the $\bar{a}$-orbit of some

$$r = \{a_1, \ldots, a_k\} \in \tbinom{\mathbb{A}}{k}.$$

Each a_i belongs to some $\bar{a}$-orbit of atoms. This orbit is an interval, so it is equal to the orbit determined by at most two atoms from $\bar{a}$. (One atom is enough if a_i belongs to $\bar{a}$, or if it is larger than (or smaller than) every atom in $\bar{a}$.) Picking these atoms for each $i = \{1, \ldots, k\}$, we obtain a tuple $\bar{b} \subseteq \bar{a}$ of at most $2k$ atoms, such that R is equal to the $\bar{b}$-orbit of r. As a consequence, R is supported by $\bar{b}$.

As a result, we may take Φ to be the set of characteristic functions of subsets of Q supported by at most $2k$ atoms. Since k is fixed for a given Q, this is an orbit-finite set.

The argument for the **equality atoms** is only a little more complicated. Here for Φ we take the set of characteristic functions of subsets of Q supported by at most k atoms.

We know that Q is the image of an equivariant surjection from $\mathbb{A}^{(k)}$ for some number k, so it is enough to deal with $Q = \mathbb{A}^{(k)}$. Let R be the $\bar{a}$-orbit of some

$$r = (a_1, \ldots, a_k) \in \mathbb{A}^{(k)}.$$

We proceed by induction on the number of atoms in r that are *not* in $\bar{a}$. If this number is zero, then R is the singleton of r and the characteristic function of R is in Φ, so there is nothing to do. If there are some atoms in r that not in $\bar{a}$, denote by $I \subseteq \{1, \ldots, k\}$ the set of coordinates where these atoms occur in r. Then an $s \in \mathbb{A}^{(k)}$ belongs to R if and only if:

(i) s is equal to r on all coordinates apart from I,
(ii) atom in s on coordinates from I do not belong to $\bar{a}$.

Let $\hat{R} \subseteq Q$ be the set of tuples s that satisfy condition (i) above; obviously $R \subseteq \hat{R}$. Moreover, $\hat{R}$ has a support of size $k - |I|$, so $\chi_{\hat{R}}$, the characteristic function of $\hat{R}$, belongs to Φ.

Let $\{r_1, \ldots, r_n\}$ be the set of all k-tuples in $\mathbb{A}^{(k)}$ that can be obtained from r by replacing some (not necessarily all, but at least one) atoms on coordinates from I by some atoms from $\bar{a}$. Let R_i be the $\bar{a}$-orbit of r_i, for each i. The sets R_i are pairwise disjoint, and their union is equal to the difference $\hat{R} \setminus R$, therefore there is a linear equation:

$$\chi_R = \chi_{\hat{R}} - \textstyle\sum_{i=1}^{n} \chi_{R_i}.$$

By the inductive assumption, each χ_{R_i} is a linear combination of functions from Φ, which completes the proof. $\Box$

Corollary VI.5 *Assume the equality or ordered atoms. If V is an orbit-finitely spanned vector space, then the same is true for its finitely supported dual.*

Proof
If a vector space is spanned by an orbit-finite set Q, then its finitely supported dual embeds into the space $Q \xrightarrow{\text{fs}} \mathbb{F}$, which is orbit-finitely spanned by Theorem VI.4. The corollary follows since orbit-finitely spanned vector spaces are closed under subspaces. $\Box$

In contrast to the finite-dimensional case, orbit-finitely spanned vector spaces are not isomorphic to their finitely supported duals. Indeed, Example 8 shows that the finitely supported dual of $\mathsf{Lin}\,\mathbb{A}$ is equivariantly isomorphic to the space $\mathsf{Lin}(1 + \mathbb{A})$. There is no linear isomorphism between them which is equivariant, or even finitely supported.

The following example shows that the restriction to equality or ordered atoms was important in Theorem VI.4. This example hints on the difficulties one may encounter when generalizing our theory to, say, arbitrary oligomorphic atom structures.

Example 9. Consider the graph atoms [3, Section 7.3.1], i.e. the case when the atoms structure $\mathbb{A}$ is the Rado graph. For an atom tuple $\bar{a}$, define $\chi_{\bar{a}} : \mathbb{A} \to \mathbb{F}$ to be the characteristic function of the common neighborhood of $\bar{a}$, i.e. the function which maps an atom to 1 if it has an edge to all atoms in $\bar{a}$, and otherwise maps the atom to 0. This is a finitely supported function, so it belongs to $\mathbb{A} \xrightarrow{\text{fs}} \mathbb{F}$. For $n \in \{1, 2, \ldots\}$, let

$$V_n \subseteq \mathbb{A} \xrightarrow{\text{fs}} \mathbb{F}$$

to be the subspace which is spanned by functions of the form $\chi_{\bar{a}}$, where $\bar{a}$ is a tuple of at most n atoms. We will show that $V_n \subsetneq V_{n+1}$ for every n.

Indeed, assume to the contrary a linear equation

$$\chi_{\bar{a}} = \sum_{i=1}^{n} p_i \cdot \chi_{\bar{b_i}}$$

where all $p_i \neq 0$ and where $\bar{a}$ strictly larger than every $\bar{b_i}$. Pick i such that $\bar{b_i}$ is minimal in the equation, i.e. such that $\bar{b_j} \not\subseteq \bar{b_i}$ for all $j \neq i$. Pick an atom c that is in the common neighborhood of $\bar{b_i}$ but not in the common neighborhood of $\bar{a}$ or any other $\bar{b_j}$; it exists by the universal property of the Rado graph. Then $\chi_{\bar{b_i}}(c) = 1$, $\chi_{\bar{a}}(c) = 0$ and $\chi_{\bar{b_j}}(c) = 0$ for $j \neq i$, therefore $p_i = 0$, a contradiction.

It follows that the vector space $\mathbb{A} \xrightarrow{\text{fs}} \mathbb{F}$ does not have finite length. We do not know if the problem is that: (a) the finite length property fails; (b) the property of being orbit-finitely spanned does not extend to dual spaces. □

VII. Minimization

Schützenberger's original paper on weighted automata contained a minimization procedure. We now describe a version of that procedure in the orbit-finite setting using the theory of orbit-finitely spanned vector spaces from Section VI.

Consider a weighted orbit-finite automaton $\mathcal{A}$ with states Q. We assume that the automaton is reachable, in the following sense: every vector in $\mathsf{Lin}\, Q$ is a linear combination of configurations corresponding to input words. For a vector $v \in \mathsf{Lin}\, Q$, define the weighted language of v to be the weighted language recognized by the automaton obtained from $\mathcal{A}$ by setting the initial map to v, i.e. the initial weight of a state is its coefficient in the vector v. Define the *syntactic congruence* $\sim$ to be the equivalence relation on $\mathsf{Lin}\, Q$ which identifies two vectors if the corresponding weighted languages are equal. It is not hard to see that $\sim$ is closed under both linear combinations and applying atom automorphisms, one can speak of an equivariant quotient vector space $(\mathsf{Lin}\, Q)/_\sim$. Equivalently, this is the quotient of the vector space $\mathsf{Lin}\, Q$ under the subspace which consists of vectors v whose corresponding weighted language is 0 everywhere. This quotient space is orbit-finitely spanned by the equivalence classes of states in Q.

In the finite dimensional case studied by Schützenberger, a minimal automaton is obtained by choosing some basis for this vector space, and using it as the state space of the minimal automaton. This idea, however, will not work in the orbit-finite setting, due to the difficulties with finding a basis that were described in Examples 6 and 7.

Example 10. For the equality atoms and the field of rational numbers, consider the weighted language $L\colon \mathbb{A}^* \to \mathbb{F}$:

$$L(w) = \begin{cases} 1 & \text{if } w = aba \text{ for some } a \neq b \in \mathbb{A}, \\ -1 & \text{if } w = abb \text{ for some } a \neq b \in \mathbb{A}, \\ 0 & \text{otherwise.} \end{cases}$$

This is recognized by a weighted orbit-finite automaton where the set of states Q is

$$\underbrace{\{\perp\}}_{\substack{\text{initial weight 1}\\ \text{final weight 0}}} + \underbrace{\mathbb{A} + \mathbb{A}^{(2)}}_{\substack{\text{initial weight 0}\\ \text{final weight 0}}} + \underbrace{\{\top\}}_{\substack{\text{initial weight 0}\\ \text{final weight 1}}}$$

and the transitions with nonzero weight are

$$\underbrace{(\perp, a, a),\ (a, b, (a, b)),\ ((a, b), a, \top)}_{\text{weight 1}},\ \underbrace{((a, b), b, \top)}_{\text{weight -1}}$$

for every $a \neq b \in \mathbb{A}$. For the syntactic congruence $\sim$, it is not hard to see that $(a, b) \sim -(b, a)$ for every $a \neq b$. The quotient space $\mathsf{Lin}\, Q/_\sim$ is:

$$\mathsf{Lin}(\, \{\perp, \top\} + \mathbb{A}\,) \ \oplus\ X,$$

where X is the space from Example 7. This space has no finitely supported basis. □

This example shows that in the orbit-finite setting the minimization procedure can leave the realm of weighted orbit-finite automata. To overcome this issue, we use an alternative model for weighted automata, which we call *orbit-finitely spanned automata*. These are deterministic automata where the state spaces are orbit-finitely spanned vector spaces and all weight functions are linear maps.

Definition VII.1 (Orbit-finitely spanned automaton) *An orbit-finitely spanned automaton consists of:*

1) *an orbit-finite input alphabet* Σ*;*
2) *an orbit-finitely spanned vector space* V*;*
3) *a finitely supported transition function*

$$\delta : V \times \Sigma \ \to\ V$$

such that $v \mapsto \delta(v, a)$ *is a linear map for every* $a \in \Sigma$*;*

4) *an initial vector* $v_0 \in V$*;*
5) *a finitely supported linear map* $F : V \to \mathbb{F}$.

An orbit-finitely spanned automaton recognizes a weighted language as expected: given an input word, it computes an element of $\mathbb{F}$ by starting in the initial vector, then applying the transition functions corresponding to the input letters, and finally applying the final function F.

Theorem VII.2 *The same weighted languages are recognized by orbit-finitely spanned automata and weighted orbit-finite automata.*

The theorem is proved in Appendix D. It differs from an analogous construction in the finite-dimensional setting in one aspect: if one converts a weighted orbit-finite automaton with states Q to an orbit-finitely spanned automaton in the natural way, then the resulting vector space $\mathsf{Lin}\, Q$ has an orbit-finite basis. Not every orbit-finitely spanned automaton arises this way though, because we do not require the vector space to have an orbit-finite basis. A key step in the proof of Theorem VII.2 is that every orbit-finitely spanned automaton is equivalent to one whose state space has a basis.

The point of orbit-finitely spanned automata is that they can be minimized. Define a *homomorphism* between orbit-finitely spanned automata $\mathcal{A}$ and $\mathcal{B}$ to be a finitely supported linear map from the state space of $\mathcal{A}$ to the state space of $\mathcal{B}$, which is consistent with the structure of the automata in the natural way, see [3, Sec. 6.2]. If $\mathcal{A}$ is an orbit-finitely spanned automaton that is reachable (i.e. its vector space is spanned by vectors that can be reached via input words), and $\sim$ is its syntactic congruence (defined in the same way as for weighted orbit-finite automata), then there is a well defined quotient automaton $\mathcal{A}_{/\sim}$. The quotient automaton admits a homomorphism from every reachable orbit-finitely spanned automaton that recognizes the same weighted language as $\mathcal{A}$. Hence, the quotient automaton is unique up to isomorphism, and can thus be called *the* minimal automaton.

We finish this section with a third perspective on weighted languages, this time phrased in terms of monoids. Define an *orbit-finitely spanned monoid* to be a monoid $(M, \cdot, 1)$ where the underlying set M is an orbit-finitely spanned vector space, and the monoid operation is bi-linear (i.e. linear in each of the two coordinates). We say that a weighted language $L\colon \Sigma^* \to \mathbb{F}$ is recognized by such a monoid if the diagram

$$\begin{tikzcd} & M \arrow[dr, "F"] & \\ \Sigma^* \arrow[ur, "h"] \arrow[rr, "L"'] & & \mathbb{F} \end{tikzcd}$$

commutes for some finitely supported monoid homomorphism h and some finitely supported linear map F. The following result is somewhat unexpected, because in the non-weighted setting, orbit-finite automata and orbit-finite monoids do not recognize the same languages [3, Exercise 91].

Theorem VII.3 *Orbit-finitely spanned monoids recognize the same weighted languages as weighted orbit-finite automata and orbit-finitely spanned automata.*

Proof

From an orbit-finitely spanned monoid we can easily construct an orbit-finitely spanned automaton, with the same underlying vector space. For the converse direction, starting with a weighted orbit-finite automaton with states Q, we build a monoid out of finitely supported functions:

$$f : (Q \times Q) \xrightarrow{\text{fs}} \mathbb{F}$$

such that for every $p \in Q$ there are finitely many states $q \in Q$ such that $f(p, q) \neq 0$. By Theorem VI.4, this vector space is orbit-finitely spanned. The monoid operation is defined by:

$$(f \cdot g)(p, q) = \sum_{r \in Q} f(p, r) \cdot g(r, q),$$

with the sum being finite by the assumption on f. This operation is finitely supported and bi-linear. The recognizing homomorphism is built using the same construction as when converting a nondeterministic automaton into a monoid. □

An advantage of the monoid approach is the symmetry between reading the input word left-to-right and right-to-left. In particular, the languages recognized by orbit-finitely spanned monoids are easily seen to be closed under reversals; this is harder to see for the remaining models.

VIII. Application to unambiguous automata

A classical application of weighted automata is a polynomial time algorithm for language equivalence of unambiguous finite automata, i.e., nondeterministic automata with at most one accepting run for every input word. Two unambiguous finite automata are equivalent (i.e. they recognize the same language) if and only if they have the same number of accepting runs for every input word (since the number of accepting runs is zero or one). For every nondeterministic finite automaton, one can easily construct in polynomial time a weighted finite automaton which maps every input word to the number of accepting runs of the nondeterministic finite automaton; and therefore two unambiguous finite automata are equivalent if and only if the corresponding weighted finite automata are equivalent.

In this section, we show how this result can be lifted from finite to orbit-finite automata. Consider first the case of unambiguous orbit-finite automata which are non-guessing, in the sense that they have finitely many initial states, and for every state q and input letter σ, there are finitely many transitions of the form (q, a, p). As explained in Example 3, for such automata we can easily count runs using a weighted orbit-finite automaton, and thus we can solve the language equivalence problem in the same time as in Theorem III.2. However, our techniques apply also to unambiguous orbit-finite automata without the non-guessing restriction.

Theorem VIII.1 *Assume that the atoms are either $(\mathbb{N}, =)$ or $(\mathbb{Q}, <)$. The equivalence problem for equivariant*[2] *unambiguous register automata*[3]*, which are allowed to use guessing, is in* ExpTime*, and in polynomial time when the number of registers is fixed.*

This improves on previous work [13], [14] in that: (a) we allow unrestricted guessing; (b) we allow ordered atoms and not just equality atoms;and (c) we improve the previous upper bounds of 2ExpSpace for an unbounded number of registers and ExpSpace for a fixed number of registers.

The rest of this section is devoted to proving the above theorem. The main observation is that an orbit-finitely spanned automaton can count accepting runs for a nondeterministic orbit-finite automaton, as stated in the following lemma.

Lemma VIII.2 *Consider an equivariant nondeterministic orbit-finite automaton $\mathcal{A}$, which has finitely many accepting runs for every input word. There is an equivariant orbit-finitely spanned automaton $\mathcal{B}$, over the field of rational numbers, which outputs for every word the number of accepting runs of $\mathcal{A}$. Furthermore, the length of the vector space used by the automaton $\mathcal{B}$ is at most*

$$2^{poly(k)} \cdot n^{O(k)}. \tag{9}$$

[2] As for Theorem III.2, this theorem would also work for finitely supported automata, but the notation would become more involved.

[3] We state the theorem using unambiguous register automata, see [13, Sect. 2] and not for general unambiguous orbit-finite automata, so that it can be more easily compared to existing results in the literature.

where k and n are as in Theorem III.2.

Proof

Let Σ be the input alphabet, and let Q be the state space of $\mathcal{A}$. Without loss of generality we assume that every state can reach some accepting state; the remaining states can be eliminated from the automaton without affecting the recognized language or the numbers of accepting runs [3, Cor. 9.12].

For an input word $w \in \Sigma^*$, define its *configuration*

$$[w] \in Q \xrightarrow{\text{fs}} \mathbb{F}$$

to be the function which maps each state q to the number of runs on w that begin in an initial state and end in state q. The configuration produces only finite numbers, because there cannot be a state q that can be reached via infinitely many runs over the same input word w; otherwise we could append some word to w and get infinitely many accepting runs. Define

$$V = \mathsf{Span}\{[w] : w \in \Sigma^*\} \subseteq Q \xrightarrow{\text{fs}} \mathbb{F}$$

to be the subspace of $Q \xrightarrow{\text{fs}} \mathbb{F}$ that is spanned by configurations. Although the definition of V uses a spanning set that is not necessarily orbit-finite (because Σ^* is not orbit-finite), the space V is orbit-finitely spanned, as an equivariant subspace of an orbit-finitely spanned vector space, see Theorem VI.3.

We use V as the state space of a orbit-finitely spanned automaton $\mathcal{B}$. Let us first prove the length bound (9). It is enough to prove the bound for the length of $Q \xrightarrow{\text{fs}} \mathbb{F}$, since V is an equivariant subspace of it. To this end, note that the set Q can be decomposed as a disjoint union of n single-orbit sets with dimension at most k. Since

$$\mathsf{length}((Q_1+Q_2)\xrightarrow{\text{fs}}\mathbb{F}) = \mathsf{length}(Q_1\xrightarrow{\text{fs}}\mathbb{F}) + \mathsf{length}(Q_2\xrightarrow{\text{fs}}\mathbb{F})$$

it is enough to show that every equivariant single-orbit set P of atom dimension at most k satisfies

$$\mathsf{length}(P \xrightarrow{\text{fs}} \mathbb{F}) \leq 2^{poly(k)}.$$

Since $P \xrightarrow{\text{fs}} \mathbb{F}$ embeds into $\mathbb{A}^{(k)} \xrightarrow{\text{fs}} \mathbb{F}$, it is enough to show that the length of the latter space is at most exponential in k. This follows from the proof of Theorem VI.4.

We now describe the remaining structure of the orbit-finitely spanned automaton. The initial state is the configuration $[\varepsilon]$, which maps initial states to 1 and non-initial states to 0. Let us now define the transition functions. For an input letter $\sigma \in \Sigma$, define a function $\delta_\sigma : V \to V$ as follows

$$\sum_{i\in I} \alpha_i[w_i] \quad \mapsto \quad \sum_{i\in I} \alpha_i[w_i\sigma].$$

We need to justify that this is well-defined. The potential problem is that the same element of V might have several decompositions as weighted sums of configurations, and the output of δ_σ should not depend on the choice of decomposition. Consider an element of V with two decompositions:

$$\sum_{w\in W} \alpha_w[w] = \sum_{w\in W} \beta_w[w],$$

for some finite set $W \subseteq \Sigma^*$ of input words, and some coefficients α_w and β_w in the field. We need to show that δ_σ produces the same output for both decomposition, i.e.

$$\sum_{w\in W} \alpha_w[w\sigma] = \sum_{w\in W} \beta_w[w\sigma]. \tag{10}$$

Both sides in (10) are functions from states to the field; and hence to prove the equality we need to show that the functions on both sides give the same output for every state $q \in Q$. Fix q. Let P be the set of states $p \in Q$ such that the automaton has a transition (p, σ, q), and furthermore p appears in the configuration of some $w \in W$ with nonzero coefficient. An important observation is that P is a finite set: because the automaton $\mathcal{A}$ has finitely many accepting runs for every input word, the set P contains finitely many states for every word in W. For every word $w \in W$ we have

$$[w\sigma](q) = \sum_{p\in P}[w](p).$$

Therefore, to prove that both functions in the equality (10) give the same value for state q, we need to show

$$\sum_{\substack{w\in W\\ p\in P}} \alpha_w[w](p) = \sum_{\substack{w\in W\\ p\in P}} \beta_w[w](p).$$

This equality is indeed true, because our assumption implies a stronger equality, namely that for every $p \in P$ we have

$$\sum_{w\in W} \alpha_w[w](p) = \sum_{w\in W} \beta_w[w](p).$$

The function δ_σ is clearly a linear map, and thus we can set the transition function of the automaton to be $\delta(v, \sigma) = \delta_\sigma(v)$. The automaton is defined so that after reading an input word w, its state is $[w]$. The final map simply takes a configuration to the sum of all coefficients that are accepting states. This concludes the proof of Lemma VIII.2. □

Proof (of Theorem VIII.1)

Consider two unambiguous register automata $\mathcal{A}_1$ and $\mathcal{A}_2$ for which we want to decide equivalence. Apply Lemma VIII.2 to each one of them, yielding orbit-finitely spanned automata $\mathcal{B}_1$ and $\mathcal{B}_2$. Using a product construction, we get another orbit-finitely spanned automaton $\mathcal{B}$ that outputs zero for words where the automata $\mathcal{A}_1$ and $\mathcal{A}_2$ agree, and nonzero for other words. The length of the vector space in $\mathcal{B}$ is at most twice the length of the vector spaces in $\mathcal{B}_1$ and $\mathcal{B}_2$, and hence it is at most

$$2^{poly(k)} \cdot n^{O(k)} \tag{11}$$

where k is the maximal number of registers used by the automata, and n is the sum of the numbers of control states. As in the proof of Theorem III.2, we conclude that the automata $\mathcal{A}_1$ and $\mathcal{A}_2$ are equivalent if and only if they are equivalent using input words and runs that use a number of atoms as bounded by (11), and the latter equivalence can be tested using Schützenberger's polynomial time algorithm for equivalence on weighted finite automata. □

References

[1] Borja Balle and Mehryar Mohri. Learning weighted automata. In *International Conference on Algebraic Informatics*, pages 1–21. Springer, 2015.

[2] Andreas Blass. Existence of bases implies the axiom of choice. *Contemporary Mathematics*, 31:31–33, 1984.

[3] Mikołaj Bojańczyk. Slightly infinite sets, 2019. Draft from September 11, 2019, https://www.mimuw.edu.pl/ bojan/upload/main-10.pdf.

[4] Mikołaj Bojańczyk, Bartek Klin, and Slawomir Lasota. Automata theory in nominal sets. *Logical Methods in Computer Science*, 10(3), 2014.

[5] Mikołaj Bojańczyk, Bartek Klin, and Slawomir Lasota. Automata theory in nominal sets. *Log. Methods Comput. Sci.*, 10(3), 2014.

[6] Mikołaj Bojańczyk and Rafal Stefanski. Single use register automata for data words. *CoRR*, abs/1907.10504, 2019.

[7] Lorenzo Clemente and Corentin Barloy. Bidimensional linear recursive sequences and universality of unambiguous register automata. In *STACS 2021, to appear*. 2021.

[8] Thomas Colcombet. Unambiguity in automata theory. In *DCFS*, volume 9118 of *Lecture Notes in Computer Science*, pages 3–18. Springer, 2015.

[9] K. R. Goodearl and R. B. Warfield, Jr. *An Introduction to Noncommutative Noetherian Rings*. London Mathematical Society Student Texts. Cambridge University Press, 2 edition, 2004.

[10] Michael Kaminski and Nissim Francez. Finite-Memory Automata. *Theor. Comput. Sci.*, 134(2):329–363, 1994.

[11] Stefan Kiefer, Andrzej S. Murawski, Joël Ouaknine, Björn Wachter, and James Worrell. On the complexity of equivalence and minimisation for q-weighted automata. *Log. Methods Comput. Sci.*, 9(1), 2013.

[12] Daniel Krob. The equality problem for rational series with multiplicities in the tropical semiring is undecidable. In W. Kuich, editor, *Automata, Languages and Programming*, pages 101–112, Berlin, Heidelberg, 1992. Springer Berlin Heidelberg.

[13] Antoine Mottet and Karin Quaas. The containment problem for unambiguous register automata. In *STACS*, volume 126 of *LIPIcs*, pages 53:1–53:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[14] Antoine Mottet and Karin Quaas. On the containment problem for unambiguous single-register automata with guessing. *CoRR*, abs/1905.12445, 2019.

[15] Masakazu Nasu and Namio Honda. Mappings induced by pgsm-mappings and some recursively unsolvable problems of finite probabilistic automata. *Information and Control*, 15(3):250–273, 1969.

[16] Frank Neven, Thomas Schwentick, and Victor Vianu. Finite state machines for strings over infinite alphabets. *ACM Trans. Comput. Log.*, 5(3):403–435, 2004.

[17] Marcel Paul Schützenberger. On the definition of a family of automata. *Information and control*, 4(2–3):245–270, 1961.

Appendix A
Proof of Lemma IV.2

For item (i), let $g : W \to W/V$ be the (equivariant) quotient map. For the "$\geq$" direction, for any chains:

$$\begin{aligned} V_1 \subsetneq \cdots \subsetneq V_n \subsetneq V \qquad & (n \text{ proper inclusions}) \\ U_1 \subsetneq \cdots \subsetneq U_m \subsetneq W/V \qquad & (m \text{ proper inclusions}) \end{aligned}$$

the chain

$$V_1 \subsetneq \cdots \subsetneq V_n \subsetneq V \subseteq g^{-1}(U_1) \subsetneq \cdots \subsetneq g^{-1}(U_m) \subsetneq W$$

has $n+m$ proper inclusions. For the "$\leq$" direction, for any chain of proper inclusions:

$$W_1 \subsetneq \cdots \subsetneq W_n \subsetneq W \tag{12}$$

consider chains:

$$W_1 \cap V \subseteq \cdots \subseteq W_n \cap V \subseteq V \tag{13}$$
$$\overrightarrow{g}(W_0) \subseteq \cdots \subseteq \overrightarrow{g}(W_n) \subseteq W/V. \tag{14}$$

All these inclusions are not necessarily proper. However, if the i-th inclusion in the first chain is an equality:

$$W_i \cap V = W_{i+1} \cap V$$

then there is some $v \in W_{i+1} \setminus W_i$ such that $v \notin V$. Since V is the kernel of g, the i-th inclusion in the second chain is then proper:

$$\overrightarrow{g}(W_i) \subsetneq \overrightarrow{g}(W_{i+1}),$$

so the length of (12) does not exceed the sum of lengths of (13) and (14).

For item (ii), apply item (i) to $W = \mathsf{Lin}(P+Q)$ and $V = \mathsf{Lin}\, P$, noting that then $W/V = \mathsf{Lin}\, Q$.

For item (iii), assume an equivariant surjection $q : P \to Q$, let $\bar{q} : \mathsf{Lin}\, P \to \mathsf{Lin}\, Q$ be the unique linear extension of q, and apply item (i) to $W = P$ and $V = \ker(\bar{q})$, noting that then $W/V = \mathsf{Lin}\, Q$.

Appendix B
Proof of Claim IV.7

Recall that a vector is in B iff it satisfies all the conditions (4). Since we consider vectors in $\mathsf{Lin}\binom{T}{k}$, only conditions (4) with $S \in \binom{T}{k-1}$ matter. There are $k \cdot \binom{n}{k-1}$ such conditions, and each of them takes the form of a linear equation over $\binom{n}{k}$ variables, with all coefficients equal to 1.

$B \cap \mathsf{Lin}\binom{T}{k}$ is therefore the null space of a 0-1 matrix with $\binom{n}{k}$ columns and $k \cdot \binom{n}{k-1}$ rows, where columns correspond to subsets of T of size k, and rows correspond to conditions (4). We will denote this matrix by $\mathbf{M}^n_k$. By the rank-nullity theorem, it is enough to prove that

$$\operatorname{rank}(\mathbf{M}^n_k) \geq \binom{n}{k} - \binom{n-k}{k}.$$

The matrix $\mathbf{M}^n_k$ is naturally divided into k equally sized parts:

$$\mathbf{M}^n_k = \begin{bmatrix} {}^{0}\mathbf{M}^n_k \\ \hline {}^{1}\mathbf{M}^n_k \\ \hline \vdots \\ \hline {}^{k-1}\mathbf{M}^n_k \end{bmatrix} \tag{15}$$

where each ${}^{i}\mathbf{M}^{n}_{k}$ stores conditions that concern the i-th orbits in the order shown in (5). Each ${}^{i}\mathbf{M}^{n}_{k}$ has $\binom{n}{k-1}$ rows and $\binom{n}{k}$ columns. The precise definition of the matrix ${}^{i}\mathbf{M}^{n}_{k}$ is that:

- its columns are indexed by subsets $R \in \binom{T}{k}$;
- its rows are indexed by subsets $S \in \binom{T}{k-1}$;
- the entry $({}^{i}\mathbf{M}^{n}_{k})_{S,R}$ is equal to 1 if $R = S \cup \{a\}$ for some atom $a \in T$ that is strictly greater than exactly i atoms in R; and it is 0 otherwise.

We will look at these component matrices more closely now. Let us begin with $i = 0$. Let a be the smallest atom in T. Arrange both the columns and the rows of ${}^{0}\mathbf{M}^{n}_{k}$ so that they are lexicographically ordered. Then the first $\binom{n-1}{k-1}$ columns and the first $\binom{n-1}{k-2}$ rows correspond to subsets of T that contain a, and the remaining $\binom{n-1}{k}$ columns and $\binom{n-1}{k-1}$ rows correspond to subsets that do not contain a. We thus identify four parts of ${}^{0}\mathbf{M}^{n}_{k}$:

$$ {}^{0}\mathbf{M}^{n}_{k} = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]. \tag{16} $$

Notice that C is a square matrix.

Looking at the definition of ${}^{0}\mathbf{M}^{n}_{k}$, we get that:

- A and B are zero matrices;
- C is the identity matrix of order $\binom{n-1}{k-1}$,
- D is equal to the matrix ${}^{0}\mathbf{M}^{n-1}_{k}$.

Altogether we obtain:

$$ {}^{0}\mathbf{M}^{n}_{k} = \left[\begin{array}{c|c} \mathbf{0} & \mathbf{0} \\ \hline \mathbf{I} & {}^{0}\mathbf{M}^{n-1}_{k} \end{array}\right]. $$

Now consider $0 < i < k$. Still focusing on a the smallest atom in T, arrange rows and columns of ${}^{i}\mathbf{M}^{n}_{k}$ as before to obtain a decomposition as in (16). Looking at the definition of ${}^{i}\mathbf{M}^{n}_{k}$, this time we get that:

- A is equal to the matrix ${}^{i-1}\mathbf{M}^{n-1}_{k-1}$,
- B and C are zero matrices,
- D is equal to the matrix ${}^{i}\mathbf{M}^{n-1}_{k}$.

Altogether we obtain:

$$ {}^{i}\mathbf{M}^{n}_{k} = \left[\begin{array}{c|c} {}^{i-1}\mathbf{M}^{n-1}_{k-1} & \mathbf{0} \\ \hline \mathbf{0} & {}^{i}\mathbf{M}^{n-1}_{k} \end{array}\right]. \tag{17} $$

Note that we applied the same arrangement of columns for all i. Coming back to (15), we therefore obtain:

$$ \mathbf{M}^{n}_{k} = \left[\begin{array}{c|c} \mathbf{0} & \mathbf{0} \\ \hline \mathbf{I} & {}^{0}\mathbf{M}^{n-1}_{k} \\ \hline {}^{0}\mathbf{M}^{n-1}_{k-1} & \mathbf{0} \\ \hline \mathbf{0} & {}^{1}\mathbf{M}^{n-1}_{k} \\ \hline {}^{1}\mathbf{M}^{n-1}_{k-1} & \mathbf{0} \\ \hline \mathbf{0} & {}^{2}\mathbf{M}^{n-1}_{k} \\ \hline \vdots & \vdots \\ \hline {}^{k-2}\mathbf{M}^{n-1}_{k-1} & \mathbf{0} \\ \hline \mathbf{0} & {}^{k-1}\mathbf{M}^{n-1}_{k} \end{array}\right] $$

Applying some rearrangement of rows, and removing some zero rows, we obtain:

$$ \mathrm{rank}(\mathbf{M}^{n}_{k}) = \mathrm{rank} \left[\begin{array}{c|c} {}^{0}\mathbf{M}^{n-1}_{k-1} & \mathbf{0} \\ \hline {}^{1}\mathbf{M}^{n-1}_{k-1} & \mathbf{0} \\ \hline \vdots & \vdots \\ \hline {}^{k-2}\mathbf{M}^{n-1}_{k-1} & \mathbf{0} \\ \hline \mathbf{I} & {}^{0}\mathbf{M}^{n-1}_{k} \\ \hline \mathbf{0} & {}^{1}\mathbf{M}^{n-1}_{k} \\ \hline \mathbf{0} & {}^{2}\mathbf{M}^{n-1}_{k} \\ \hline \vdots & \vdots \\ \hline \mathbf{0} & {}^{k-1}\mathbf{M}^{n-1}_{k} \end{array}\right] \tag{18} $$

Now we use the general fact that

$$ \mathrm{rank} \left[\begin{array}{c|c} A & B \\ \hline \mathbf{0} & C \end{array}\right] \geq \mathrm{rank}(A) + \mathrm{rank}(C) $$

to conclude

$$ \mathrm{rank}(\mathbf{M}^{n}_{k}) \geq \mathrm{rank} \left[\begin{array}{c} {}^{0}\mathbf{M}^{n-1}_{k-1} \\ \hline {}^{1}\mathbf{M}^{n-1}_{k-1} \\ \hline \vdots \\ \hline {}^{k-2}\mathbf{M}^{n-1}_{k-1} \\ \hline \mathbf{I} \end{array}\right] + \mathrm{rank} \left[\begin{array}{c} {}^{1}\mathbf{M}^{n-1}_{k} \\ \hline {}^{2}\mathbf{M}^{n-1}_{k} \\ \hline \vdots \\ \hline {}^{k-1}\mathbf{M}^{n-1}_{k} \end{array}\right]. \tag{19} $$

Thanks to the identity component, the first summand above equals $\binom{n-1}{k-1}$. Let us now calculate the rank:

$$ R^{n}_{k} = \mathrm{rank} \left[\begin{array}{c} {}^{1}\mathbf{M}^{n}_{k} \\ \hline {}^{2}\mathbf{M}^{n}_{k} \\ \hline \vdots \\ \hline {}^{k-1}\mathbf{M}^{n}_{k} \end{array}\right]. $$

This is similar to the matrix $\mathbf{M}^{n}_{k}$, but without the 0'th component. Use (17) again and rearrange rows to obtain:

$$ R^{n}_{k} = \mathrm{rank} \left[\begin{array}{c|c} {}^{0}\mathbf{M}^{n-1}_{k-1} & \mathbf{0} \\ \hline {}^{1}\mathbf{M}^{n-1}_{k-1} & \mathbf{0} \\ \hline \vdots & \vdots \\ \hline {}^{k-2}\mathbf{M}^{n-1}_{k-1} & \mathbf{0} \\ \hline \mathbf{0} & {}^{1}\mathbf{M}^{n-1}_{k} \\ \hline \mathbf{0} & {}^{2}\mathbf{M}^{n-1}_{k} \\ \hline \vdots & \vdots \\ \hline \mathbf{0} & {}^{k-1}\mathbf{M}^{n-1}_{k} \end{array}\right] = \mathrm{rank}(\mathbf{M}^{n-1}_{k-1}) + R^{n-1}_{k}. $$

This entails

$$ R^{n}_{k} = \textstyle\sum_{m=k-1}^{n-1} \mathrm{rank}(\mathbf{M}^{m}_{k-1}). $$

Combining this with (19) we obtain:

$$ \mathrm{rank}(\mathbf{M}^{n}_{k}) \geq \binom{n-1}{k-1} + R^{n-1}_{k} = \binom{n-1}{k-1} + \textstyle\sum_{m=k-1}^{n-2} \mathrm{rank}(\mathbf{M}^{m}_{k-1}). \tag{20} $$

There is also the base case $\mathrm{rank}(\mathbf{M}^{k}_{k}) = 1$ since the matrix $\mathbf{M}^{k}_{k}$ has only one column (and k nonzero rows). Another base case is $\mathrm{rank}(\mathbf{M}_1^n) = 1$, since the matrix $\mathbf{M}_1^n$ has only one row (filled with n ones). Denote $S_k^n = \mathrm{rank}(\mathbf{M}_k^n)$ and pretend that the inequality in (20) is an equality; we then obtain the recurrence:

$$\begin{aligned}
&S_k^k = 1\\
&S_1^n = 1\\
&S_k^n = \tbinom{n-1}{k-1} + \textstyle\sum_{m=k-1}^{n-2} S_{k-1}^m \quad \text{for } n > k > 1.
\end{aligned}$$

The formula

$$S_k^n = \tbinom{n}{k} - \tbinom{n-k}{k}$$

satisfies the recurrence. Indeed, calculate:

$$\begin{aligned}
&\tbinom{n-1}{k-1} + \textstyle\sum_{m=k-1}^{n-2} \left(\tbinom{m}{k-1} - \tbinom{m-k+1}{k-1}\right) =\\
&= \tbinom{n-1}{k-1} + \textstyle\sum_{m=n-k}^{n-2} \tbinom{m}{k-1} =\\
&= \textstyle\sum_{m=n-k}^{n-1} \tbinom{m}{k-1} = \tbinom{n}{k} - \tbinom{n-k}{k},
\end{aligned}$$

where the last equality follows from a repeated application of Pascal's rule:

$$\tbinom{m}{k-1} + \tbinom{m}{k} = \tbinom{m+1}{k}.$$

This proves that $S_k^n \geq \binom{n}{k} - \binom{n-k}{k}$ and completes the proof of Claim IV.7.

Appendix C
Proof of Lemma IV.10

Define the $\bar{a}$-length of a vector space to be the maximal length of $\bar{a}$-supported subspaces. In the special case when $\bar{a}$ is the empty tuple, we get the notion of length from Definition IV.1. Every $\bar{a}$-supported orbit-finite set can be obtained, using images under $\bar{a}$-supported functions and disjoint unions, from $\bar{a}$-orbits contained in $\mathbb{A}^k$. Since Lemma IV.2 holds, with the same proof, for $\bar{a}$-length, it remains to show that the $\bar{a}$-length is finite for $\mathsf{Lin}\, Q$ when Q is a single $\bar{a}$-orbit contained in $\mathbb{A}^k$. We now split into two proofs, depending on whether we deal with the equality or ordered atoms.

- *Equality atoms.* Choose some bijection

  $$f : \underbrace{(\mathbb{A} - \bar{a})}_{\substack{\text{atoms that do}\\ \text{not appear in } \bar{a}}} \to \mathbb{A},$$

  which is possible since both sets are countably infinite. (Note that f cannot be finitely supported.) Let $\ell \in \{1, \ldots, k\}$ be the number of coordinates that are not from $\bar{a}$ in some (equivalently, every) tuple from the $\bar{a}$-orbit Q. We can lift f to an injective function (in fact, a bijection)

  $$g : Q \to \mathbb{A}^{(\ell)},$$

  which erases the coordinates that use atoms from $\bar{a}$ and applies f to the remaining coordinates. One can easily see that

  $$\mathsf{Lin}\, g : \mathsf{Lin}\, Q \to \mathsf{Lin}\, \mathbb{A}^{(\ell)}$$

  maps $\bar{a}$-supported subspaces of $\mathsf{Lin}\, Q$ to equivariant subspaces of $\mathsf{Lin}\, \mathbb{A}^{(\ell)}$, and preserves strict inclusions. Therefore the $\bar{a}$-length of $\mathsf{Lin}\, Q$ is at most the (equivariant) length of $\mathsf{Lin}\, \mathbb{A}^{(\ell)}$, and the latter length is finite by Lemma IV.9.
- *Ordered atoms.* Let the atoms in $\bar{a}$ be

  $$a_1 < \cdots < a_n.$$

  For every $i \in \{0, 1, \ldots, n\}$, consider the interval

  $$\mathbb{A}_i = \{a : a_i < a < a_{i+1}\},$$

  where a_0 is $-\infty$ and a_{n+1} is ∞. Choose some order-preserving bijection

  $$f_i : \mathbb{A}_i \to \mathbb{A}.$$

  As in the previous item, let ℓ be the coordinates from Q which avoid atoms from $\bar{a}$. By erasing the coordinates which use atoms from $\bar{a}$, and applying the appropriate functions f_i to the remaining coordinates, we get an injective function

  $$g : Q \to \mathbb{A}^{\ell}.$$

  Since the functions f_i all have the same co-domain, namely $\mathbb{A}$, the image of the function will contain tuples that are not necessarily strictly increasing. Nevertheless, the linear lifting

  $$g : \mathsf{Lin}\, Q \to \mathsf{Lin}\, \mathbb{A}^{\ell}$$

  maps $\bar{a}$-supported subspaces in $\mathsf{Lin}\, Q$ to equivariant subspaces in $\mathsf{Lin}\, \mathbb{A}^{\ell}$, and hence we obtain a finite bound on $\bar{a}$-supported chains in $\mathsf{Lin}\, Q$.

Appendix D
Equivalence of linear and weighted automata

We begin with the easier inclusion.

Lemma D.1 *For every weighted orbit-finite automaton $\mathcal{W}$ there is an orbit-finitely spanned automaton $\mathcal{L}$ which recognizes the same weighted language.*

Proof
If Q are the states of $\mathcal{W}$, then the vector space of $\mathcal{L}$ is $\mathsf{Lin}\, Q$, and the transition function

$$\delta : \mathsf{Lin}\, Q \times \mathbb{A} \to \mathsf{Lin}\, Q$$

is defined so that for every input letter $a \in \Sigma$ and states $q, p \in Q$ (which are basis vectors of the vector space $\mathsf{Lin}\, Q$), the coefficient for state p in the vector $\delta(q, a)$ is the weight of the transition (q, a, p). In other words, linear map $\delta(\_, a)$ is defined by a $Q \times Q$ matrix where the coefficient in cell (q, p) is the weight of the transition (q, a, p). Note how condition (*) from Definition III.1 is used to ensure that the $\delta(q, a)$ is a well-defined vector, in the sense that it has finitely many nonzero coordinates. The initial vector of $\mathcal{L}$ consists of the initial weights in $\mathcal{W}$, and the final function is the linear extension of the original final weight function. □

Example 11. Recall the weighted orbit-finite automaton from Example 10. We have already described the state space of the corresponding orbit-finitely spanned automaton:

$$V \;=\; \mathsf{Lin}(\{\bot,\top\}+\mathbb{A}) \;\oplus\; X,$$

and we will now define the rest od it. We define the transition function on the generators of the state space:

$$\delta(\bot,a)=a \quad \delta(\top,a)=0 \quad \delta(a,b)=(a,b)-(b,a)$$
$$\delta((a,b)-(b,a),c)=\begin{cases}\top & \text{if } c=a,\\ -\top & \text{if } c=b,\\ 0 & \text{otherwise.}\end{cases}$$

Formally, one has to show that δ is well-defined on X, i.e. it satisfies $\delta((a,b)-(b,a),c)=-\delta((b,a)-(a,b),c)$. Similarly for the final function, we define it for generators:

$$F(\bot)=0 \quad F(\top)=1 \quad F(a)=0 \quad F((a,b)-(b,a))=0.$$

The initial vector is simply $v_0=\bot$. This automaton accepts the same language L and is in fact minimal. □

Call an orbit-finitely spanned automaton *basic* if its state vector space has a basis. For a basic automaton an equivalent weighted orbit-finite automaton can be easily produced, by using the basis as the states. Therefore, to complete the proof of Theorem VII.2, we prove the following:

Lemma D.2 *For every orbit-finitely spanned automaton, there is a basic one that recognizes the same weighted language.*

Proof
Consider an orbit-finitely spanned automaton $\mathcal{A}$ with a state space V is spanned by an orbit-finite set Q. Define a *polynomial orbit-finite set*, see [6, Definition 1], to be any set which is a finite disjoint union of sets of the form $\mathbb{A}^k$. As for every orbit-finite set, there exists a polynomial orbit-finite set P with a surjective finitely supported function from P to Q. Extend this function to surjective a linear map

$$h : \mathsf{Lin}\, P \to V.$$

We will define a orbit-finitely spanned automaton $\mathcal{B}$ with vector space $\mathsf{Lin}\, P$ so that h becomes a homomorphism of orbit-finitely spanned automata, that is: a finitely supported linear map between the underlying vector spaces, which is consistent with the initial states, transition functions and final functions in the expected way, see [3, Sec. 6.2]. If two orbit-finitely spanned automata are connected by a homomorphism, then they recognize the same weighted language. Therefore, to prove the lemma it remains to define the initial state, transition function and final function in $\mathcal{B}$ so that h is a homomorphism.

For the initial state in $\mathcal{B}$ we choose some vector that is mapped by h to the initial state of $\mathcal{A}$, and for the final function we use the composition of h and the final function of $\mathcal{A}$. The transition function is defined using the following claim.

Claim D.3 *There is a finitely supported function γ which makes the following diagram commute*

$$\begin{tikzcd} P \times \Sigma \arrow[r, "\gamma"] \arrow[d, "(h,id)"'] & \mathsf{Lin}\, P \arrow[d, "h"] \\ V \times \Sigma \arrow[r, "\textit{transition function of } \mathcal{A}"'] & V \end{tikzcd}$$

Proof
Consider the composition of the following relations: the function (h, id), the transition function of $\mathcal{A}$, and the inverse of h. This is a finitely supported binary relation

$$R \subseteq (P \times \Sigma) \times \mathsf{Lin}\, P$$

such that every element of $P \times \Sigma$ is related with at least one element of $\mathsf{Lin}\, P$ (thanks to the surjectivity of h). By the Uniformization Lemma from [6, Lemma 20], there exists a finitely supported function γ which is contained in R, thus proving the lemma. It is worth pointing out that the Uniformization Lemma changes supports: even if R is equivariant, it could be the case that γ needs non-empty support. □

Extend, using linearity, the function γ from the claim to a finitely supported function

$$\bar{\gamma} : \mathsf{Lin}\, P \times \Sigma \to \mathsf{Lin}\, P$$

which is a linear map for every fixed input letter; the resulting function can then be used as the transition function for $\mathcal{B}$. The commuting diagram in the claim ensures that h is a homomorphism. □