Functional Closure Properties of Finite $\mathbb{N}$-Weighted Automata

Julian Dörfler
Saarland Informatics Campus (SIC), Saarbrücken Graduate School of Computer Science, Saarland University, Germany

Christian Ikenmeyer
University of Warwick, Coventry, UK

Abstract

We determine all functional closure properties of finite $\mathbb{N}$-weighted automata, even all multivariate ones, and in particular all multivariate polynomials. We also determine all univariate closure properties in the promise setting, and all multivariate closure properties under certain assumptions on the promise, in particular we determine all multivariate closure properties where the output vector lies on a monotone algebraic graph variety.

2012 ACM Subject Classification Theory of computation → Automata extensions

Keywords and phrases Finite automata, weighted automata, counting, closure properties, algebraic varieties

Digital Object Identifier 10.4230/LIPIcs.ICALP.2024.134

Category Track B: Automata, Logic, Semantics, and Theory of Programming

Related Version *Full Version*: https://arxiv.org/abs/2404.14245

Funding *Christian Ikenmeyer*: EPSRC EP/W014882/2

1 Finite $\mathbb{N}$-weighted automata and functional closure properties

Let Σ be a finite set, for example $\Sigma = \{0, 1\}$. A finite $\mathbb{N}$-weighted automaton with all weights 1 is a nondeterministic finite automaton that on input $w \in \Sigma^\star$ outputs the number of accepting computation paths on input w, instead of just outputting whether or not an accepting computation path exists, see Def. 2.1 for the formal definition [1]. While every nondeterministic finite automaton determines a subset of $\Sigma^\star$, a finite $\mathbb{N}$-weighted automaton computes a function $\Sigma^\star \to \mathbb{N}$. A function $f : \Sigma^\star \to \mathbb{N}$ can be presented as the series $\sum_{w \in \Sigma^\star} f(w)w$, and the set of series is denoted by $\mathbb{N}\langle\langle\Sigma^\star\rangle\rangle$ in the automata literature, see e.g. [9][2]. The natural way of adding two functions $\Sigma^\star \to \mathbb{N}$ and adding two series in $\mathbb{N}\langle\langle\Sigma^\star\rangle\rangle$ coincides, but in both presentations we have a natural way of taking the product, and those do not coincide:

1. Pointwise product of functions $\Sigma^\star \to \mathbb{N}$. This is called the Hadamard product.
2. Convolution of series, called the Cauchy product.

A series f is called *recognizable* if there is a finite $\mathbb{N}$-weighted automaton that computes f. The set of recognizable series is denoted by $\mathbb{N}^{\mathrm{rec}}\langle\langle\Sigma^\star\rangle\rangle$ in [9], but we denote it by $\#\mathsf{FA}$, to emphasise that we undertake a study similar to $\#\mathsf{P}$ in [11, Thm 3.13], [4, Thm 6], and recently [12], but instead of polynomial-time Turing machines we study finite automata.

[1] We use the equality of the number of accepting paths of an NFA and the output of the corresponding $\mathbb{N}$-weighted automaton, see [9, Exa. 2.2].

[2] A series with finite support is called a *polynomial*, but we will not be concerned with the support of series in this paper. Instead, we use the term *polynomial* as it is used in commutative algebra, and we mean multivariate polynomials with rational coefficients.

51st International Colloquium on Automata, Languages, and Programming (ICALP 2024).
Editors: Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson;
Article No. 134; pp. 134:1–134:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The Kleene-Schützenberger theorem states that #FA is the smallest set that contains all support 1 series and is closed under sums, Cauchy products, and Kleene-iterations (whenever well-defined, a Kleene-iteration is the sum of all Cauchy powers), see [9, §4], but we will not need this insight.

In this paper we study the functional closure properties[3] of #FA. A function $\varphi : \mathbb{N}^m \to \mathbb{N}$ is called a *functional closure property* of #FA if for all $f_1 \in \#\mathsf{FA}$, $f_2 \in \#\mathsf{FA}$, …, $f_m \in \#\mathsf{FA}$ we have that $\varphi(f_1, \dots, f_m) \in \#\mathsf{FA}$. By $\varphi(f_1, \dots, f_m)$ we mean the function that on input $w \in \Sigma^\star$ outputs $\varphi(f_1(w), \dots, f_m(w))$.

Classically, one of the simplest functional closure properties of #FA is $\varphi : \mathbb{N}^2 \to \mathbb{N}$, $\varphi(f_1, f_2) = f_1 + f_2$. This is a functional closure property of #FA, because given $f_1 \in \#\mathsf{FA}$ and $f_2 \in \#\mathsf{FA}$, we can show $\varphi(f_1, f_2) = f_1 + f_2 \in \#\mathsf{FA}$ by an easy construction: The new NFA consists of a copy of the NFA for f_1 and a copy of the NFA for f_2, and makes an initial nondeterministic choice as to which NFA to run, see Lemma 3.1 for the details.

Another classical simple functional closure property of #FA is $\varphi : \mathbb{N}^2 \to \mathbb{N}$, $\varphi(f_1, f_2) = f_1 \cdot f_2$. This corresponds to the Hadamard product. This is a functional closure property of #FA, because given $f_1 \in \#\mathsf{FA}$ and $f_2 \in \#\mathsf{FA}$, we can show $\varphi(f_1, f_2) = f_1 \cdot f_2 \in \#\mathsf{FA}$ by the following construction: The new NFA consists of the product NFA of the NFAs for f_1 and f_2, and the accepting states correspond to pairs of accepting states, see Lemma 3.2 for the details. This product construction corresponds to the Hadamard product.

The Cauchy product is also a product on the set #FA, but we explain now that the Cauchy product is not "functional", and hence it is out of scope for this type of studies. If $\varphi : \mathbb{N}^m \to \mathbb{N}$ is a functional closure property of #FA, then we can study the corresponding map $\widetilde{\varphi} : \underbrace{\#\mathsf{FA} \times \#\mathsf{FA} \times \cdots \times \#\mathsf{FA}}_{m \text{ times}} \to \#\mathsf{FA}$. Observe that if φ is a functional closure property of #FA, then by definition we have that for all pairs $(w, w') \in \Sigma^\star \times \Sigma^\star$:

$$\text{if } (f_1(w), \dots, f_m(w)) = (f_1(w'), \dots, f_m(w')), \text{ then } \widetilde{\varphi}(f_1, \dots, f_m)(w) = \widetilde{\varphi}(f_1, \dots, f_m)(w').$$

Let $\zeta : \#\mathsf{FA} \to \#\mathsf{FA}$ denote the Cauchy square. We use the observation above to show that ζ is not equal to $\widetilde{\varphi}$ for any $\varphi : \mathbb{N} \to \mathbb{N}$. Let $m = 1$ and $f(w) = 1$ if $w = 1$, $f(w) = 0$ otherwise. Clearly, $f \in \#\mathsf{FA}$. Then $\zeta(f)(11) = 1$, and $\zeta(f)(w) = 0$ for all $w \neq 11$. In particular $\zeta(f)(0) \neq \zeta(f)(11)$, even though $f(0) = f(11)$. Hence, $\zeta \neq \widetilde{\varphi}$ for all $\varphi : \mathbb{N} \to \mathbb{N}$.

Numerous functional closure properties of #FA exist, for example the safe decrementation $\max\{0, f_1 - 1\}$, and the binomial coefficient $\binom{f_1}{2}$. But not all non-negative functions are functional closure properties of #FA, for example $(f_1 - f_2)^2$ is not, which can be shown using the Pumping Lemma. In this paper, we determine *all* functional closure properties of #FA, see §1.2 for the detailed statement.

1.1 Motivation

Functional closure properties can be studied for many different counting machine models (also for example with different types of oracle access) and different types of input sets. The first study of this type was done for nondeterministic polynomial-time Turing machines, i.e., the class #P, see [11], [4], and the recent [12]. Recall that the class #P is the class of functions $f : \Sigma^* \to \mathbb{N}$ for which a nondeterministic polynomial time Turing machine M exists such that for all $w \in \Sigma^*$ the number of accepting paths for the computation $M(w)$ is exactly $f(w)$. The papers mentioned above prove that the relativizing multivariate polynomial

[3] See [11, Sec. 1] for the naming *functional closure property*. A different reasonable name would be *pointwise closure property*.

closure properties are exactly those polynomials that have nonnegative integers in their expansion over the binomial basis, see [12]. A functional closure property $\varphi : \mathbb{N}^m \to \mathbb{N}$ of $\#\mathsf{P}$ is *relativizing* if φ is a closure property for all $\#\mathsf{P}^A$, where $A \subset \Sigma^*$ is some oracle. The hope is that for simpler models of computation no oracle access is required to determine the functional closure properties, and we show that this is true for $\#\mathsf{FA}$, see §1.2.

Functional closure properties can be used directly to construct combinatorial proofs of equalities and inequalities. For example, Fermat's little theorem states that p divides $a^p - a$. The quantity $\frac{1}{p}(a^p - a)$ has a combinatorial interpretation, which can be deduced from the fact that $\frac{1}{p}((f_1)^p - f_1)$ is a univariate functional closure property of $\#\mathsf{P}$, see [12, Prop. 7.3.1], which coincides with the original proof [18], see also [17, eq. (5)]). On the other hand, if a function is not a functional closure property, then this means in a very strong sense that there is no combinatorial interpretation for the quantity it describes. For example, the Hadamard inequality ([10, §2.13], [3, §2.11], [12, eq. (2)]) states that

$$\det\begin{pmatrix} a_{11} & \cdots & a_{1d} \\ \vdots & \ddots & \vdots \\ a_{d1} & \cdots & a_{dd} \end{pmatrix}^2 \leq \prod_{i=1}^d (a_{i1}^2 + \cdots + a_{id}^2).$$

One could try to prove this by finding a combinatorial interpretation of the difference $\mathcal{H} \geq 0$ of the right-hand side and the left-hand side, but even for $d = 3$ we have that

$$\varphi(f_1, \ldots, f_9) \;=\; (f_1^2 + f_2^2 + f_3^3) \cdot (f_4^2 + f_5^2 + f_6^3) \cdot (f_7^2 + f_8^2 + f_9^3) - \det\begin{pmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{pmatrix}$$

is not a 9-variate relativizing functional closure property of $\#\mathsf{P}$, see [12, §7.2]. In particular, if the function is not a closure property of $\#\mathsf{P}$, then there are instantiations $f_1, \ldots, f_9 \in \#\mathsf{P}$ such that $\varphi(f_1, \ldots, f_9)$ is not in $\#\mathsf{P}$, whereas a combinatorial interpretation of $\mathcal{H}$ should yield $\varphi(f_1, \ldots, f_9) \in \#\mathsf{P}$. This does not rule out a more indirect combinatorial proof for the inequality: For example, for proving combinatorially that $(a-1)^2 \geq 0$ one could try to interpret the quantity $(a-1)^2$ combinatorially, but $(f_1 - 1)^2$ is not a relativizing closure property of $\#\mathsf{P}$. However, $f_1 \cdot (f_1 - 1)^2 = 6\binom{f_1}{3} + 2\binom{f_1}{2}$ is a relativizing closure property of $\#\mathsf{P}$ (see [12, §2.4]). There is an obvious combinatorial interpretation of $6\binom{a}{3} + 2\binom{a}{2}$ as counting size 2 and 3 subsets with multiplicity 6 and 2, respectively. Hence this gives an indirect combinatorial proof for the inequality $(a-1)^2 \geq 0$ by providing a combinatorial interpretation for $a(a-1)^2$.

Some inequalities are only true if the inputs satisfy certain constraints. For example, the *Ahlswede Daykin inequality*, see [1], [2], [12, §1.2(3)]: If $a_0 b_0 \geq c_0 d_0$ and $a_0 b_1 \geq c_0 d_1$ and $a_1 b_0 \geq c_0 d_1$ and $a_1 b_1 \geq c_1 d_1$, then $(c_0 + c_1)(d_0 + d_1) \geq (a_0 + a_1)(b_0 + b_1)$. If all quantities are in $\#\mathsf{P}$, including the differences $c_0 d_0 - a_0 b_0$, can we conclude that $(c_0 + c_1)(d_0 + d_1) - (a_0 + a_1)(b_0 + b_1)$ is in $\#\mathsf{P}$? This is an example of a promise problem: We are given twelve $\#\mathsf{P}$ functions $a_0, a_1, b_0, b_1, c_0, c_1, d_0, d_1, h_1, h_2, h_3, h_4$ with the guarantee that $a_0 b_0 + h_1 = c_0 d_0$, $a_0 b_1 + h_2 = c_0 d_1$, $a_1 b_0 + h_3 = c_0 d_1$, $a_1 b_1 + h_4 = c_1 d_1$. In other words, the 12-dimensional output vector that we get for every $w \in \Sigma^*$ lies on a codimension 4 algebraic subvariety in $\mathbb{Q}^{12}$. Recall that an algebraic subvariety is defined as the simultaneous zero set of a set of polynomials. Since the 4 variables $h_1, \ldots, h_4$ are determined by the other 8, this variety is a so-called *graph* or *graph variety*. Numerous questions about combinatorial proofs for inequalities from different areas of mathematics can be phrased in the language of graph varieties, see [12]. The idea is to collect the equations for a set S (the variety) into what is called the *vanishing ideal* I, i.e., $I = I(S) = \{\varphi \in \mathbb{Q}[f_1, \ldots, f_m] \mid \forall (f_1, \ldots, f_m) \in S : \varphi(f_1, \ldots, f_m) = 0\}$; and define the *coordinate ring* $\mathbb{Q}[S]$ as the quotient ring $\mathbb{Q}[f_1, \ldots, f_m]/I(S)$, see [7]. An element in the quotient ring is a coset with respect to the vanishing ideal. If there exists a representative

φ' in a coset $\varphi + I$ that is a functional closure property of #P, then every function in $\varphi + I$ is a promise closure property of #P on the variety S. It is desirable to also have the opposite direction, but this only holds under some reasonable restrictions on S, in particular it holds for all graph varieties. This is used in [12, Prop. 2.5.1] to show that $c_0 d_0 - a_0 b_0$ is not a relativizing promise closure property of #P on this graph variety. We prove the same strong dichotomy for monotone graph varieties for #FA instead of #P, see Theorem 4.9.

The systematic study of combinatorial interpretations and combinatorial proofs via definitions from computational complexity theory is a very recent research direction [15, 16, 12, 17, 13, 5, 6]. The goal is to determine whether or not certain quantities admit a combinatorial description or not. Famous open questions of this type in algebraic combinatorics have been listed by Stanley in [22], for example his problems 9, 10, and 12. As many combinatorialists do, Stanley has phrased his questions in an informal way without mentioning counting classes.

The class #P is the correct class for some purposes, but for others it is too large. For example, the determinant of a skew-symmetric matrix with entries from $\{-1, 0, 1\}$ is always non-negative, but this quantity is trivially in #P, because the determinant can be computed in polynomial time. This gives no satisfying insight into whether or not this quantity has a combinatorial interpretation. Smaller counting classes are required (see also the discussion in [15, §1]), and we provide the first study of functional closure properties for the subclass #FA ⊂ #P. Unlike the classification for #P, our results do not rely on oracle separations, i.e., our classification is entirely unconditional.

1.2 Our results

Let $n \operatorname{rem} p \in \{0, \ldots, p-1\}$ denote the smallest nonnegative r such that $n \equiv_p r$. A function $\varphi : \mathbb{N} \to \mathbb{N}$ is called *ultimately PORC* (Polynomial On Residue Classes[4]) if $\exists p, N \in \mathbb{N}$ and there exist polynomials $\varphi_0, \ldots, \varphi_{p-1} : \mathbb{N} \to \mathbb{Q}$ such that for every $n \geq N$ we have $\varphi(n) = \varphi_{n \operatorname{rem} p}(n)$ [5].

We first classify the univariate functional closure properties of #FA:

- **Theorem** (see Theorem 3.22). *A function $\varphi : \mathbb{N} \to \mathbb{N}$ is a functional closure property of* #FA *if and only if φ is an ultimately PORC function.*

More generally, we classify the multivariate functional closure properties of #FA:

- **Theorem** (see Theorem 3.23). *A function $\varphi : \mathbb{N}^m \to \mathbb{N}$ is a functional closure property of* #FA *if and only if φ can be written as a finite sum of finite products of univariate ultimately PORC functions.*

We analyze the special case of multivariate polynomials:

- **Theorem** (see Lemma 3.26). *A multivariate polynomial $\varphi : \mathbb{N}^m \to \mathbb{N}$ with rational coefficients is a functional closure property of* #FA *iff for every ψ that can be formed from φ by replacing any subset of variables – including the empty set – by constants from $\mathbb{N}$, then all dominating terms of ψ in the binomial basis have positive coefficients.*

[4] PORC functions are also known as quasipolynomials or pseudopolynomials, but we want to avoid those names for the potential confusion to quasipolynomial growth and pseudopolynomial running times.

[5] Note that each φ in this paper is defined on the natural numbers and maps to the natural numbers, which is a subtle restriction. For example, a univariate polynomial $\varphi : \mathbb{Q} \to \mathbb{Q}$ maps integers to integers if and only if its coefficients in the binomial basis are integers, see Section 2. However, non-negativity is not an algebraic property. Also note that for the case of φ being just a univariate polynomial, the corresponding linear recursive sequence can have negative entries in the matrix.

We lift this result to monotone graph varieties (and to more general sets, see Theorem 4.6), where we get exactly the desirable classification given by the vanishing ideal:

- **Theorem** (see Theorem 4.9). *Let S be a monotone graph variety and let $I = I(S)$ be its vanishing ideal. A multivariate polynomial $\varphi : S \to \mathbb{N}$ is a functional promise closure property of #FA with regard to S if and only if there exists $\psi \in I$ such that $\varphi + \psi$ is a multivariate functional closure property of #FA.*

2 Notation

Let $\mathbb{N} = \{0, 1, 2, \ldots\}$. For a finite set Σ let $\Sigma^\star$ denote the set of all finite length sequences with elements from Σ. The vector space of multivariate polynomials $\mathbb{Q}[f_1, \ldots, f_m]$ in variables $f_1, \ldots, f_m$ has a basis given by products of binomial coefficients: $\left\{ \prod_{i=1}^{m} \binom{f_i}{c_i} \right\}_{c_1,\ldots,c_m}$, where each $c_i \in \mathbb{N}$. Here we used $\binom{x}{c} = \frac{1}{c!} x \cdot (x-1) \cdot \ldots \cdot (x-c+1)$ as a polynomial. This is called the *binomial basis*. A multivariate polynomial φ is called *integer valued* if $\varphi(\mathbb{Z}^m) \subseteq \mathbb{Z}$, which is equivalent to $\varphi(\mathbb{N}^m) \subseteq \mathbb{Z}$, and which is also equivalent to all coefficients in the binomial basis being integers, see for example [12, Prop. 4.2.1] for a short proof of this classical fact.

We now recall (see [9, Def. 2.1]) our main model of computation, the finite $\mathbb{N}$-weighted automaton, which we just call non-deterministic finite automaton (NFA) for brevity.

▶ **Definition 2.1.** *An NFA M is a tuple $(Q, \Sigma, \mathrm{wt}, \mathrm{in}, \mathrm{out})$ where the set of states Q and the alphabet Σ are finite sets and $\mathrm{wt} : Q \times \Sigma \times Q \to \mathbb{N}$ is the weighted transition function, $\mathrm{in} : Q \to \mathbb{N}$ are the weighted initial states and $\mathrm{out} : Q \to \mathbb{N}$ are the weighted accepting states*[6]. *A* computation *P for a word $w = w_1 \ldots w_n \in \Sigma^\star$ of length n is a sequence $q_0 q_1 \ldots q_n$ in Q^{n+1}. It has* multiplicity *or* weight[7] $\mathbf{w}(P) = \mathrm{in}(q_0) \cdot \prod_{i=1}^{n} \mathrm{wt}(q_{i-1}, w_i, q_i) \cdot \mathrm{out}(q_n)$ *and* partial weight $\underline{\mathbf{w}}(P) = \mathrm{in}(q_0) \cdot \prod_{i=1}^{n} \mathrm{wt}(q_{i-1}, w_i, q_i)$. *We say that M* computes $f : \Sigma^\star \to \mathbb{N}$ *where $f(w)$ is the sum of the weights over all computations of M on w. The class #FA is defined as the set of all functions $f : \Sigma^\star \to \mathbb{N}$ that are computed by NFAs.*

If needed to distinguish these for different automata, we use a corresponding subscript, for example the weights of computations in M_f would be denoted by $\mathbf{w}_f$, etc.

▶ **Definition 2.2** (Simple NFA). *We say an NFA $M = (Q, \Sigma, \mathrm{wt}, \mathrm{in}, \mathrm{out})$ is* simple *if* $\mathrm{im}\,\mathrm{wt}, \mathrm{im}\,\mathrm{in}, \mathrm{im}\,\mathrm{out} \subseteq \{0, 1\}$.

The notion of a simple NFA also motivates our use of the term NFA opposed to $\mathbb{N}$-weighted automaton: We simply count the number of accepting paths of M on a word w. This is in line with #P counting the number of accepting paths on a polynomial time non-deterministic Turing machine.

▶ **Lemma 2.3** (Folklore). *For every NFA M there exists a simple NFA M' computing the same function.*

---

[6] Note that in definitions by other authors one can find simpler versions of NFAs, in particular unweighted initial and accepting states and unweighted edges while also restricting to a single initial state. We will see soon that working with unweighted NFAs is not a restriction, but additionally restricting the model to have a single initial state is strictly weaker, since this model could not compute any function f with $f(\varepsilon) > 1$. To obtain the same expressiveness one would have to additionally allow for ε-transitions, while disallowing cycles of ε-transitions to prevent infinite values for f.

[7] We will use both of these terms interchangeably. For a weighted automaton, calling this weight is more natural, while when looking at the underlying graph as a multigraph, multiplicity of paths and walks is more natural.

A proof of this simple fact can be found in the appendix of the full version, for the sake of completeness. We denote by $a \equiv_p b$ that $a \in \mathbb{N}$ and $b \in \mathbb{N}$ are congruent modulo $p \in \mathbb{N} \setminus \{0\}$. The indicator function $\mathbb{1}_{n=c} : \mathbb{N} \to \mathbb{N}$ is defined as $n \mapsto \begin{cases} 1 & \text{if } n = c \\ 0 & \text{otherwise} \end{cases}$ and analogously for different conditions. By abuse of notation, if we have a function $f : \Sigma^\star \to \mathbb{N}$ and an expression $\varphi : \mathbb{N} \to \mathbb{N}$ in n, we replace n by f in the expression to denote $\varphi \circ f$. For example we use $\mathbb{1}_{f=c}$ to denote the function $w \mapsto \mathbb{1}_{f(w)=c}$, similarly $\binom{f}{2}$ denotes the function $w \mapsto \binom{f(w)}{2}$. Furthermore we use the notation $[n]$ to denote the set $\{1, \ldots, n\}$ for any $n \in \mathbb{N}$.

3 Functional closure properties

3.1 Univariate functional closure properties

We say a function $\varphi : \mathbb{N} \to \mathbb{N}$ is a functional closure property of #FA if $\varphi(\#\mathsf{FA}) \subseteq \#\mathsf{FA}$, i.e. if for every function $f \in \#\mathsf{FA}$ the function $\varphi \circ f$ is also in #FA. Our goal in this section is to classify all functional closure properties of #FA. They will be precisely the ultimately PORC functions.

We call a function $\varphi : \mathbb{N} \to \mathbb{N}$ an *ultimately almost PORC function* if there is a *quasiperiod* p, an *offset* $N \in \mathbb{N}$ and *constituents* $\varphi_0, \ldots, \varphi_{p-1} : \mathbb{N} \to \mathbb{Q}$, where each φ_i is either a polynomial with rational coefficients or a function in $2^{\Theta(n)}$, and for every $n \geq N$ we have $\varphi(n) = \varphi_{n \operatorname{rem} p}(n)$. If all the constituents are polynomials, we call φ an *ultimately PORC function.* The smallest representative of each constituent and of the finite cases before the periodic behaviour is captured by the *shifted remainder* operator $n \operatorname{srem}_p N$, defined via

$$n \operatorname{srem}_p N = \begin{cases} n & \text{if } n < N \\ \min\{k \geq N \mid k \equiv_p n\} & \text{if } n \geq N \end{cases}$$

The first half of the section is dedicated to proving that every ultimately PORC function is a functional closure property of #FA, see Lemma 3.18. In order to prove this we show that #FA is closed under

▶ **Lemma 3.1** (Addition). *If $f, g \in \#\mathsf{FA}$, then $f + g \in \#\mathsf{FA}$.*

▶ **Lemma 3.2** (Multiplication). *If $f, g \in \#\mathsf{FA}$, then $f \cdot g \in \#\mathsf{FA}$.*

▶ **Lemma 3.9** (Subtraction of constants). *If $f \in \#\mathsf{FA}$, then $\forall c \in \mathbb{N} :\ \max(f - c, 0) \in \#\mathsf{FA}$.*

▶ **Lemma 3.10** (Clamping). *If $f \in \#\mathsf{FA}$, then $\min(f, c) \in \#\mathsf{FA}$ for any constant $c \in \mathbb{N}$.*

▶ **Lemma 3.11** (Comparison with constants). *If $f \in \#\mathsf{FA}$, then the functions $\mathbb{1}_{f=c}$, $\mathbb{1}_{f \leq c}$, $\mathbb{1}_{f \geq c}$ are in #FA for any constant $c \in \mathbb{N}$.*

▶ **Lemma 3.13** (Division by constants). *If $f \in \#\mathsf{FA}$, then $\forall c \in \mathbb{N} \setminus \{0\} :\ \lfloor f/c \rfloor \in \#\mathsf{FA}$.*

▶ **Lemma 3.14** (Modular arithmetic). *If $f \in \#\mathsf{FA}$, then the function $\mathbb{1}_{f \equiv_c d}$ is in #FA for any constants $c \in \mathbb{N} \setminus \{0\}$ and $d \in \mathbb{Z}_c$.*

▶ **Lemma 3.15** (Binomial coefficients). *If $f \in \#\mathsf{FA}$, then $\binom{f}{c} \in \#\mathsf{FA}$ for any constant $c \in \mathbb{N}$.*

While addition and multiplication are technically bivariate functional closure properties, we list them here already since they are abundantly used throughout the proofs of the univariate functional closure properties. Proofs of those two classical results can be found in [8, Ch. 4.1 and 4.2.2] and in the appendix of the full version, for the sake of completeness.

With the exception of binomial coefficients all the other closure properties need to be able to “remove” some of the possible computations. For example, consider decrementation, the special case of truncated subtraction by one, and consider some simple NFA M computing some strictly positive function f. We now want to construct an NFA M' that computes $f-1$, i.e. an NFA that has exactly one non-zero computation less than M (assuming computations of weights zero or one). For this we want a procedure to single out one non-zero computation of M to then change its weight to zero. For stronger models of computation – like polynomial time non-deterministic Turing machines – this approach seems hopeless. Already deciding the existence of one such computation is NP-hard. However for NFAs, deciding the existence of a non-zero computation can be decided by a deterministic finite automaton, namely the powerset automaton. Adjusting the powerset construction to filter out a single non-zero computation, namely the lexicographically minimal one can then be used to show that decrementation is a closure property of #FA.

Generalizing this approach to more general properties about the computations gives us the framework of stepwise computation properties:

▶ **Definition 3.3** (Stepwise computation property). *Let $M = (Q, \Sigma, \mathrm{wt}, \mathrm{in}, \mathrm{out})$ be an NFA. A* stepwise computation property prop *is defined as* $\mathrm{prop} = (S, \mathrm{init}, \mathrm{step}, \mathrm{cond})$ *where S is a finite set and* $\mathrm{init} : Q \to S$, $\mathrm{step} : Q \times \Sigma \times Q \times S \to S$ *and* $\mathrm{cond} : S \to \{0, 1\}$ *are functions. For $w = w_1 \dots w_n \in \Sigma^\star$ and a computation $P = q_0 \dots q_n$ of M on w we define a* step sequence $s_0 := \mathrm{init}(q_0)$ *and* $s_i := \mathrm{step}(q_{i-1}, w_i, q_i, s_{i-1})$ *for $i \in [n]$. We also write* $\mathrm{prop}(w, P) := \mathrm{cond}(s_n)$ *to be the evaluation of the property.*

These stepwise computation properties now enable us, given a simple NFA, to construct NFAs computing both of the following:

▶ **Lemma 3.4.** *Let M_f be a simple NFA computing a function f and let* prop *be a stepwise computation property. Then there is an NFA M computing $g(w) = \sum_P \mathbf{w}_f(P) \cdot \mathrm{prop}(w, P)$, where the sum is over all computations P of M_f on w.*

▶ **Lemma 3.5.** *Let M_f be a simple NFA computing a function f and let* prop *be a stepwise computation property. Then there is an NFA M computing $g(w) = \sum_P \mathrm{prop}(w, P)$, where the sum is over all computations P of M_f on w.*

Proof sketch. For both of these lemmas, we construct a sort of product automaton of M_f and prop (represented by the set S), the details can be found in the appendix of the full version. ◀

In other words, stepwise computation properties allow us to either “disable” specific computations of M_f or they allow us to directly extract information about the computations of M_f. Note that the restriction on the finiteness of S is necessary, as the elements of S are hard-coded into the state space of the NFA in Lemmas 3.4 and 3.5. In particular, these lemmas do not hold for even countably infinite S, which can be seen with the example $S = \mathbb{N}$ when we define the stepwise computation property in such a way that $\mathrm{prop}(w, P) = 1$ iff $|w|$ is prime, leading to an NFA recognizing the language of all words of prime length, a well known contradiction.

Further note that these two constructions do not incur exponential blowups themselves, however for most of our applications the set S will be of exponential size in the number of states of M_f.

Returning to our decrementation example, to use Lemma 3.4 we want to construct a stepwise computation property prop with $\mathrm{prop}(w, P) = 0$ iff P is the lexicographically smallest non-zero weight computation on w. For this we can set $S = \mathcal{P}(Q)$ to be the set of all subsets of states Q, denoting the set of states that currently are the endpoints of lexicographically smaller partial computations of non-zero weight than the partial computation P we are on. We need to store all such potential states, since some current partial non-zero weight computations might not be possible to be completed to a full non-zero weight computation. Initially this set contains all states that are smaller than the start state of P, the step function then checks which lexicographically smaller partial computations can be extended and whether any new partial computations that agreed with P up to this state can be lexicographically smaller than P. Finally the cond function then checks whether there are any such lexicographically smaller partial computations left that can be completed to a non-zero weight computation, i.e. that end on a state $q \in Q$ with $\mathrm{out}(q) = 1$.

We want to generalize this idea to be able to generally create stepwise computation properties that argue about the number of non-zero computations, either in total or lexicographically smaller than a given computation. However doing this in general would require choosing the set S as the set of functions $Q \to \mathbb{N}$, which is infinite. As a result we embed the number of computations into finite semirings first to then extract the relevant information. For this purpose we only consider semirings with both additive and multiplicative identities. Homomorphisms h from a semiring $\mathcal{R}$ into another semiring $\mathcal{R}'$ need to fulfill $h(a + b) = h(a) + h(b)$, $h(a \cdot b) = h(a) \cdot h(b)$, $h(1_{\mathcal{R}}) = 1_{\mathcal{R}'}$, $h(0_{\mathcal{R}}) = 0_{\mathcal{R}'}$. Since every element of $\mathbb{N}$ is either 0 or can be formed by repeated addition of 1, any homomorphism from $\mathbb{N}$ into any other semiring is uniquely defined.

We can then use $\mathcal{R}$ to construct stepwise computation properties (the full proofs can be found in the appendix of the full version). Combined with Lemmas 3.4 and 3.5 these use similar ideas to [14].

▶ **Lemma 3.6.** *Let $N = (Q, \Sigma, \mathrm{wt}, \mathrm{in}, \mathrm{out})$ be a simple NFA and let $\mathcal{R}$ be a finite semiring and let $\tau : \mathbb{N} \to \mathcal{R}$ be the unique homomorphism from $\mathbb{N}$ to $\mathcal{R}$. For any function $\pi : \mathcal{R} \to \{0, 1\}$ there is a stepwise computation property* prop *with* $\mathrm{prop}(w, P) = \pi(\tau(\sum_{P'} \mathbf{w}(P')))$*, where the sum is over all computations P' of N on w, independent of P.*

Proof sketch. Construct prop $= (S, \mathrm{init}, \mathrm{step}, \mathrm{cond})$ via: $S = Q \to \mathcal{R}$, $\mathrm{init}(q) = r \mapsto \tau(\mathrm{in}(r))$, $\mathrm{step}(q, \sigma, q', s) = r \mapsto \sum_{r' \in Q} s(r') \cdot \tau(\mathrm{wt}(r', \sigma, r))$, and $\mathrm{cond}(s) = \pi\big(\sum_{r \in Q} s(r) \cdot \tau(\mathrm{out}(r))\big)$. Let $s_0, \ldots, s_n$ be the step sequence of any computation P of w. Since τ is a homomorphism, we can pull out τ. Thus $s_i(q) = \tau(\sum_{\tilde{P}} \underline{\mathbf{w}}(\tilde{P}))$, where the sum is over all computations $\tilde{P}$ of $w_1, \ldots w_i$ ending in the state q. The condition cond then completes this to $\mathrm{prop}(w, P) = \pi(\tau(\sum_{P'} \mathbf{w}(P')))$, where the sum is over all computations P' of N on w. ◀

▶ **Lemma 3.7.** *Let $M = (Q, \Sigma, \mathrm{wt}, \mathrm{in}, \mathrm{out})$ be a simple NFA with some ordering $<$ of Q, let $\mathcal{R}$ be a finite semiring and let $\tau : \mathbb{N} \to \mathcal{R}$ be the unique homomorphism from $\mathbb{N}$ to $\mathcal{R}$. For any function $\pi : \mathcal{R} \to \{0, 1\}$ there is a stepwise computation property* prop *with* $\mathrm{prop}(w, P) = \pi(\tau(\sum_{P'} \mathbf{w}(P')))$ *for all non-zero computations P of N on w, where the sum is over all computations P' of N on w that are lexicographically smaller than P.*

Proof sketch. Construct prop $= (S, \mathrm{init}, \mathrm{step}, \mathrm{cond})$ via: $S = Q \to \mathcal{R}$, $\mathrm{init}(q) = r \mapsto \tau(\mathbb{1}_{r<q} \cdot \mathrm{in}(r))$, $\mathrm{step}(q, \sigma, q', s) = r \mapsto \sum_{r' \in Q} s(r') \cdot \tau(\mathrm{wt}(r', \sigma, r)) + \tau(\mathbb{1}_{r<q'} \cdot \mathrm{wt}(q, \sigma, r))$, $\mathrm{cond}(s) = \pi(\sum_{r \in Q} \tau(\mathrm{out}(r)) \cdot s(r))$. Let $s_0, \ldots, s_n$ be the step sequence of any computation $P = q_0 \ldots q_n$ of w. Inductively we can show that $s_i(r) = \sum_{P'} \tau(\underline{\mathbf{w}}(P'))$ for all $i \in \{0, \ldots, n\}$ and $r \in Q$, where the sum is over all computations $P' = q'_0 \ldots q'_i$ of N on $w_1 \ldots w_i$ with

$q'_i = r$ that are lexicographically smaller than $q_0 \dots q_i$. Initially the only computations $P' = q'_0$ that are lexicographically smaller than the computation q_0 are the ones with $q'_0 < q_0$. For $i > 0$ for a computation $P' = q'_0 \dots q'_i$ to be lexicographically smaller than $q_0 \dots q_i$ there are two possibilities. Either $q'_0 \dots q'_{i-1}$ is already lexicographically smaller than $q_0 \dots q_{i-1}$ or $q'_0 \dots q'_{i-1} = q_0 \dots q_{i-1}$ and $q'_i < q_i$. In the second case the weight of P' is precisely $w(q_{i-1}, w_i, q'_i)$ since P is a computation of non-zero weight and thus weight exactly 1. Combining all of this, we can finish the proof of the claim with a similar argument to Lemma 3.6. ◀

Note that the previous lemma makes no statement about the value of $\mathrm{prop}(w, P)$ for any computations P of weight zero. However, this is enough for our uses, since we only combine it with Lemma 3.4, i.e., $\mathrm{prop}(w, P)$ gets weighted by $\mathbf{w}(P)$.

Most commonly, as is the case for decrementation, we want to be able to exactly distinguish the number of non-zero computations if it is less than k and otherwise be able to tell that the number is at least k. This is achieved by using the following capped semiring:

▶ **Definition 3.8** (Capped semiring). *For $k \in \mathbb{N}$ we call the semiring $\mathcal{R}_k = \{0, \dots, k\}$ with the operations $a +_{\mathcal{R}} b := \min(a + b, k)$ and $a \cdot_{\mathcal{R}} b := \min(a \cdot b, k)$ the* capped *semiring.*

We can now show that decrementation is a functional closure property by simply using Lemma 3.7 using the capped semiring $\mathcal{R}_1$ and $\pi(a) = a$ to construct our wanted stepwise computation property computing $\mathrm{prop}(w, P) = 0$ iff P is the lexicographically smallest non-zero weight computation on w.

Most of the remaining closure properties are now proven by using the capped semiring of a specific size and choosing the function π accordingly, we will show this in detail for the example of subtraction, the remaining proofs can be found in the appendix of the full version.

▶ **Lemma 3.9** (Subtraction of constants). *If $f \in \#\mathsf{FA}$, then $\forall c \in \mathbb{N} : \max(f - c, 0) \in \#\mathsf{FA}$.*

Proof. Let $M_f = (Q_f, \Sigma, \mathrm{wt}_f, \mathrm{in}_f, \mathrm{out}_f)$ be a simple NFA computing f with an arbitrary ordering $<$ on Q_f. Lemma 3.7 on the capped semiring $\mathcal{R}_c$ with $\pi(a) = \mathbb{1}_{a \geq c}$ for all $a \in \mathcal{R}$ constructs a stepwise computation property prop with

$$\mathrm{prop}(w, P) = \begin{cases} 1 & \text{if the number of non-zero computations } P' \text{ on } w \text{ that arelex.} \\ & \text{smaller than } P \text{ is at least } c \\ 0 & \text{otherwise} \end{cases}$$

for all computations P on w of non-zero weight, i.e., $\mathrm{prop}(w, P) = 0$ iff P is one of the c lexicographically smallest computations on w with non-zero weight. It follows that the NFA M constructed by Lemma 3.4 computes $g(w) = \max(f(w) - c, 0)$. ◀

If instead of rejecting the c lexicographically smallest computations, we accept only those computations, we compute the minimum of f and c.

▶ **Lemma 3.10** (Clamping). *If $f \in \#\mathsf{FA}$, then $\min(f, c) \in \#\mathsf{FA}$ for any constant $c \in \mathbb{N}$.*

By using the capped semiring $\mathcal{R}_{c+1}$ with $\pi_=(a) = \mathbb{1}_{a=c}$, $\pi_\leq(a) = \mathbb{1}_{a \leq c}$ and $\pi_\geq(a) = \mathbb{1}_{a \geq c}$, we can compute the indicator functions $\mathbb{1}_{f=c}$, $\mathbb{1}_{f \leq c}$ and $\mathbb{1}_{f \geq c}$ respectively.

▶ **Lemma 3.11** (Comparison with constants). *If $f \in \#\mathsf{FA}$, then the functions $\mathbb{1}_{f=c}$, $\mathbb{1}_{f \leq c}$, $\mathbb{1}_{f \geq c}$ are in $\#\mathsf{FA}$ for any constant $c \in \mathbb{N}$.*

The previous lemma in particular also implies the following:

▶ **Lemma 3.12.** *If $\varphi : \mathbb{N} \to \mathbb{N}$ is a functional closure property of* #FA *and $\psi : \mathbb{N} \to \mathbb{N}$ is an arbitrary function with $\varphi(n) = \psi(n)$ for all but finitely many $n \in \mathbb{N}$, then ψ is also a functional closure property of* #FA*.*

Proof. Let $f \in$ #FA be arbitrary. Further let $N \in \mathbb{N}$ be such that $\varphi(n) = \psi(n)$ for all $n \geq N$. Then $(\psi \circ f)(w) = \sum_{i=0}^{N-1} \mathbb{1}_{f(w)=i}\psi(i) + \mathbb{1}_{f(w)\geq N}\varphi(f(w))$ for all $w \in \Sigma^\star$. In particular $\psi \circ f \in$ #FA since the $\psi(i)$ are constants and $\varphi \circ f \in$ #FA. ◀

Division and modular arithmetic however use a different semiring: they use the finite cyclic semiring $\mathbb{Z}_c = \{0, \dots, c-1\}$ with $\pi(a) = \mathbb{1}_{a=c-1}$ and $\pi(a) = \mathbb{1}_{a=d}$ respectively.

▶ **Lemma 3.13** (Division by constants)**.** *If $f \in$ #FA, then $\forall c \in \mathbb{N} \setminus \{0\} : \lfloor f/c \rfloor \in$ #FA.*

▶ **Lemma 3.14** (Modular arithmetic)**.** *If $f \in$ #FA, then the function $\mathbb{1}_{f \equiv_c d}$ is in* #FA *for any constants $c \in \mathbb{N} \setminus \{0\}$ and $d \in \mathbb{Z}_c$.*

The previous closure properties turn out to already be sufficient to generate all functional closure properties, so in particular they are sufficient to generate binomial coefficients by using subtraction of constants, multiplication and division by constants by using the definition of binomial coefficients as a polynomial: $\binom{x}{c} = \frac{1}{c!} x \cdot (x-1) \cdot \ldots \cdot (x-c+1)$. Nonetheless, we give an additional proof for binomial coefficients as a different interesting application of the stepwise computation property framework.

▶ **Lemma 3.15** (Binomial coefficients)**.** *If $f \in$ #FA, then $\binom{f}{c} \in$ #FA for any constant $c \in \mathbb{N}$.*

Proof. Let $M_f = (Q_f, \Sigma, \mathrm{wt}_f, \mathrm{in}_f, \mathrm{out}_f)$ be a simple NFA computing f and let $c \in \mathbb{N}$. For $c < 2$ the statement of this lemma is trivially true, so assume $c \geq 2$.

We construct the c-fold product automaton $M_f^c = (Q_f^c, \Sigma, \mathrm{wt}_f^c, \mathrm{in}_f^c, \mathrm{out}_f^c)$ with

$$\begin{aligned}
\mathrm{wt}_f^c((q_1, \dots, q_c), \sigma, (q_1', \dots, q_c')) &= \textstyle\prod_{i=1}^c \mathrm{wt}_f(q_i, \sigma, q_i') \\
\mathrm{in}_f^c((q_1, \dots, q_c)) &= \textstyle\prod_{i=1}^c \mathrm{in}_f(q_i) \\
\mathrm{out}_f^c((q_1, \dots, q_c)) &= \textstyle\prod_{i=1}^c \mathrm{out}_f(q_i)
\end{aligned}$$

M_f^c is a simple NFA and every computation on M_f^c is the cartesian product of c computations on M_f. Our aim is to now construct a stepwise computation property $\mathrm{prop} = (S, \mathrm{init}, \mathrm{step}, \mathrm{cond})$ such that $\mathrm{prop}(w, P) = 1$ iff P is composed of c pairwise distinct computations[8] on N_f.

For this let S be the set of all equivalence relations on the set $[c]$. We define $\mathrm{init}((q_0, \dots, q_c))$ to be the equivalence relation R_0 with $(a, b) \in R_0$ iff $q_a = q_b$. Additionally we define $\mathrm{cond}(R) = 1$ iff R is the equivalence relation where every element is only equivalent to itself, i.e. a computation gets accepted iff all its constituent computations are pairwise distinct. Finally we define $\mathrm{step}((q_1, \dots, q_c), \sigma, (q_1', \dots, q_c'), R)$ to be the equivalence relation R' defined via $(a, b) \in R'$ iff $(a, b) \in R$ and $q_a' = q_b'$. With a simple induction we can prove that for a computation $P = P_1 \times \ldots \times P_c$ and the step sequence $R_0, \dots, R_n$ we have $(a, b) \in R_i$ iff the computations P_a and P_b are identical for the first i steps. It follows that the NFA M constructed by Lemma 3.4 computes $g(w) = \binom{f(w)}{c} \cdot c!$. Now, $\binom{f}{c} \in$ #FA by Lemma 3.13. ◀

---

[8] We could also require them to be sorted in lexicographical order by having S be the set of all total preorders, but since we can divide by $c!$ we are going with the easier exposition.

While the combination of the previous lemmas can be used to show that any polynomial written in the binomial basis with non-negative integer coefficients is a functional closure property of #FA, we can do better by considering a shifted binomial basis. For example, consider the polynomial $\varphi(x) = \frac{x^2}{2} - \frac{3x}{2} + 1$. This polynomial is non-negative for all $x \in \mathbb{N}$. Writing φ in the binomial basis we get $\varphi(x) = \binom{x}{2} - \binom{x}{1} + 1$. If however we allow the upper indices of the binomial basis to be shifted, we can write φ without the use of negative coefficients as $\varphi(x) = \binom{x-1}{2}$. While $x-1$ itself is not a functional closure property of #FA the function $\max(x-1, 0)$ is a functional closure property of #FA and is different from $x-1$ for only finitely many $x \in \mathbb{N}$. In the same way we see that $\varphi'(x) := \binom{\max(x-1,0)}{2}$ only differs from φ for finitely many $x \in \mathbb{N}$, namely $x = 0$. Using Lemma 3.12 to change those finitely many values, we see that φ is indeed a functional closure property of #FA.

Generalizing this idea we will show with the next two lemmas that this is possible for any φ with integer coefficients in the binomial basis, with a small restriction: We don't show that φ itself is a functional closure property of #FA, but rather that $x \mapsto \max(\varphi(x), 0)$ is one. Note that this restriction is the best we can hope for, since no computation in an NFA can ever have negative weight.

▶ **Lemma 3.16.** *Let $\varphi(x) = \sum_{i=0}^{r} a_i \cdot \binom{x}{i}$ with $a_i \in \mathbb{Z}$ and $a_r > 0$. Then there are $b_0, \ldots, b_r \in \mathbb{N}$ and $c_0, \ldots, c_r \in \mathbb{N}$ with $\varphi(x) = \sum_{i=0}^{r} b_i \cdot \binom{x-c_i}{i}$.*

Proof sketch. We inductively prove this claim by using the Chu-Vandermonde identity [21] on the term of highest degree. It allows us to replace the highest degree binomial via $\binom{x-c_r}{r} = \sum_{i=0}^{r} (-1)^{r-i} \binom{r-i+c_r-1}{r-i} \binom{x}{i}$. For sufficiently large $c_r \in \mathbb{N}$ this implies that the leading term of $\varphi(x) - a_r \cdot \binom{x-c_r}{r}$ again is positive and of smaller degree. ◀

A full proof of the previous lemma can be found in the appendix of the full version.

▶ **Lemma 3.17** (Integer-valued polynomials)**.** *Let $f \in$ #FA and let $\varphi : \mathbb{Q} \to \mathbb{Q}$ be an integer-valued polynomial, then $\max(\varphi \circ f, 0) \in$ #FA.*

Proof. We can assume the leading coefficient of φ to be positive. Otherwise $\max(\varphi \circ f, 0)$ can be directly written as a finite sum $\sum_i c_i \cdot \mathbb{1}_{f=i}$ which is in #FA by Lemmas 3.1, 3.2 and 3.11. Write φ in the binomial basis as $\varphi(x) = a_0 \cdot \binom{x}{0} + \ldots + a_r \cdot \binom{x}{r}$ with $a_r > 0$. Since φ is integer-valued, all of the a_i are integers, see [12, Prop. 4.2.1]. Using Lemma 3.16 we get a representation $\varphi(x) = \sum_{i=0}^{r} b_i \cdot \binom{x-c_i}{i}$ with $b_0, \ldots, b_r \in \mathbb{N}$ and $c_0, \ldots, c_r \in \mathbb{N}$. For $x \geq \max\{c_i \mid 0 \leq i \leq r\} =: N$ we have that $\binom{x-c_i}{i} = \binom{\max(x-c_i,0)}{i}$ and thus $\psi(x) := \sum_{i=0}^{r} b_i \cdot \binom{\max(x-c_i,0)}{i}$ only differs from $x \mapsto \max(\varphi(x), 0)$ on finitely many inputs and is a functional closure property of #FA by Lemmas 3.1, 3.2, 3.9 and 3.15. Lemma 3.12 then finishes off the claim. ◀

We now have the tools available to show our claim that every ultimately PORC function is a functional closure property of #FA.

▶ **Lemma 3.18.** *Every ultimately PORC function is a functional closure property of #FA.*

Proof. Let $\varphi : \mathbb{N} \to \mathbb{N}$ be an ultimately PORC function with period p comprised of the polynomial constituents $\varphi_0, \ldots, \varphi_{p-1} : \mathbb{N} \to \mathbb{Q}$ and $N \in \mathbb{N}$, s.t. for all $n \geq N$ we have $\varphi(n) = \varphi_{n \operatorname{rem} p}(n)$. Additionally let $f \in$ #FA. We can write

$$\varphi \circ f = \textstyle\sum_{i=0}^{N-1} \mathbb{1}_{f=i} \cdot \varphi(i) + \mathbb{1}_{f \geq N} \cdot \left(\sum_{i=0}^{p-1} \mathbb{1}_{f \equiv_p i} \cdot \lfloor \max(\varphi_i \circ f, 0) \rfloor\right).$$

Figure 1 NFAs computing the functions $1^n \mapsto n$ and $1^n \mapsto 2^n$ respectively. Edges with a multiplicity of 2 are denoted by listing the edge label twice.

Combining Lemmas 3.1, 3.2, 3.11 and 3.14 this shows $\varphi \circ f \in \#\mathsf{FA}$, if we can show $\lfloor \max(\varphi_i \circ f, 0) \rfloor \in \#\mathsf{FA}$ for all $i \in \{0, \ldots, p-1\}$. To show this let α_i be the common denominator of the coefficients of φ_i. Then $\alpha_i \cdot \varphi_i$ is a polynomial with integer coefficients, so it in particular is an integer-valued polynomial and by Lemma 3.17 we have that $\max(\alpha_i \cdot \varphi_i \circ f, 0) \in \#\mathsf{FA}$. Combining this with Lemma 3.13 we get that $\left\lfloor \frac{\max(\alpha_i \cdot \varphi_i \circ f, 0)}{\alpha_i} \right\rfloor = \lfloor \max(\varphi_i \circ f, 0) \rfloor \in \#\mathsf{FA}$. ◀

The remainder of this section is dedicated to showing that no other functional closure properties of #FA exist. This will make use of the following well known algebraic interpretation of NFAs:

▶ **Lemma 3.19** (see [19]). *If $M = (Q, \Sigma, \mathrm{wt}, \mathrm{in}, \mathrm{out})$ is an NFA, then there are matrices $A_\sigma \in \mathbb{N}^{|Q| \times |Q|}$ for each symbol $\sigma \in \Sigma$ and vectors $a, b \in \mathbb{N}^{|Q|}$, s.t. M computes $a^T \cdot \left(\prod_{j=1}^{|w|} A_{w_j}\right) \cdot b$ for all $w \in \Sigma^\star$.*

Proof. We index A_σ, a and b using states $q, q' \in Q$. Choose $(A_\sigma)_{q,q'} = \mathrm{wt}(q, \sigma, q')$, $a_q = \mathrm{in}(q)$ and $b_q = \mathrm{out}(q)$. It is now easy to see that M computes exactly $a^T \cdot \left(\prod_{j=1}^{|w|} A_{w_j}\right) \cdot b$. ◀

When restricting to a unary alphabet $\Sigma = \{\sigma\}$, this degenerates the computed function to $a^T \cdot A_\sigma^{|w|} \cdot b$. In order to analyze the behaviour of these functions we first analyze the behaviour of the matrix power as a function in $|w|$ in the next two lemmas. Their proofs can be found in the appendix of the full version.

▶ **Lemma 3.20.** *Let $A \in \mathbb{N}^{k \times k}$. Then any diagonal entry of A^n is a function $f : \mathbb{N} \to \mathbb{N}$ with one of the following properties:*
1. *$f(n) = 0$ for all $n \in \mathbb{N} \setminus \{0\}$ and $f(0) = 1$.*
2. *There is a $p \in \mathbb{N} \setminus \{0\}$, such that for all $n \in \mathbb{N}$ we have $f(n) = \mathbb{1}_{n \equiv_p 0}$.*
3. *There is a $p \in \mathbb{N} \setminus \{0\}$ and a function $g \in 2^{\Theta(n)}$, such that for all $n \in \mathbb{N}$ we have $f(n) = \mathbb{1}_{n \equiv_p 0} \cdot g(n)$.*

These naturally correspond to vertices v in the multigraph defined by the adjacency matrix A with
1. no paths from v to v.
2. exactly one path from v to v of length p.
3. multiple walks from v to v where the lengths of all the walks from v to v have gcd p.

We can then lift this result to all entries of A^n.

▶ **Lemma 3.21.** *If $A \in \mathbb{N}^{k \times k}$, then each entry of A^n is an ultimately almost PORC function.*

▶ **Theorem 3.22** (Classification of univariate functional closure properties of #FA). *A function $\varphi : \mathbb{N} \to \mathbb{N}$ is a functional closure property of #FA iff φ is an ultimately PORC function. This even holds when #FA is restricted to unary languages.*

Proof. Lemma 3.18 already shows that every ultimately PORC function is a closure property of #FA. It remains to show that all functional closure properties of #FA are ultimately PORC functions. For this let $\varphi : \mathbb{N} \to \mathbb{N}$ be a functional closure property of #FA. The function

$f : \{1\}^\star \to \mathbb{N}$ defined by $f(1^n) = n$ is computed by the left NFA in Figure 1 and thus in $\#\mathsf{FA}$. Consequently $\varphi \circ f \in \#\mathsf{FA}$. Let $M = (Q, \{1\}, \mathrm{wt}, \mathrm{in}, \mathrm{out})$ be an NFA computing $\varphi \circ f$. By Lemma 3.19 this NFA induces a transition matrix $A \in \mathbb{N}^{|Q|\times|Q|}$ and vectors $a, b \in \mathbb{N}^{|Q|}$, s.t. $a^T A^n b = \varphi(f(1^n)) = \varphi(n)$ for all $n \in \mathbb{N}$. Every entry of A^n is an ultimately almost PORC function by Lemma 3.21 and thus $a^T A^n b = \varphi(n)$ is one as well. Let p be the quasiperiod of φ and let φ_i be one of the constituents of φ corresponding to some residue class $i \in \{0, \dots, p-1\}$. Assume for the sake of contradiction that φ_i grows in $2^{\Theta(n)}$, i.e. there is a constant $\gamma \in \mathbb{R}^+$ and $N \in \mathbb{N}$, s.t. for every $n \geq N$ we have $\varphi_i(n) \geq 2^{\gamma n}$. Consider the function $f_{p,i} : \{1\}^\star \to \mathbb{N}$ defined by $f_{p,i}(1^n) = p \cdot (2^n + N) + i$. We claim $f_{p,i}$ is in $\#\mathsf{FA}$. The function $1^n \mapsto 2^n$ is computed by the right NFA in Figure 1 and thus in $\#\mathsf{FA}$. The remainder of the claim follows by Lemma 3.1 and Lemma 3.2. Note that $f_{p,i}(1^n) \equiv_p i$, so $\varphi \circ f_{p,i} = \varphi_i \circ f_{p,i}$ has to be in $\#\mathsf{FA}$ as well. Furthermore $\varphi(f_{p,i}(1^n)) = \varphi_i(f_{p,i}(1^n)) = \varphi_i(p \cdot (2^n + N) + i) \geq 2^{\gamma p \cdot (2^n+N)+\gamma i}$ for all $n \in \mathbb{N}$ which is larger than any NFA can compute, since NFAs can only compute functions that are at most linearly exponential in the length of the input. We conclude that none of the constituents φ_i of φ can be exponential, so they are instead all polynomials, making φ an ultimately PORC function. ◀

3.2 Multivariate functional closure properties

▶ **Theorem 3.23** (Classification of multivariate functional closure properties of $\#\mathsf{FA}$). *A function $\varphi : \mathbb{N}^m \to \mathbb{N}$ is a functional closure property of $\#\mathsf{FA}$ iff φ can be written as a finite sum of finite products of univariate ultimately PORC functions.*

Proof. By Theorem 3.22 any univariate ultimately PORC function is a closure property of $\#\mathsf{FA}$. As such any finite sum or finite product of them is also a closure property of $\#\mathsf{FA}$ by Lemmas 3.1 and 3.2.

It remains to show that all functional closure properties of $\#\mathsf{FA}$ are of this form. For this let $\varphi : \mathbb{N}^m \to \mathbb{N}$ be a functional closure property of $\#\mathsf{FA}$. Define the alphabet $\Sigma = \{\sigma_1, \dots, \sigma_m\}$ and the functions $f_i : \Sigma^\star \to \mathbb{N}$ where $f_i(w) := \#_i(w)$ is defined as the number of occurences $\#_i(w)$ of the symbol σ_i in w. Applying the closure property to $f_1, \dots, f_m$ gives that $\varphi \circ (f_1, \dots, f_m) \in \#\mathsf{FA}$ and thus is computed by an NFA $M = (Q, \Sigma, \mathrm{wt}, \mathrm{in}, \mathrm{out})$. This induces transition matrices $A_\sigma \in \mathbb{N}^{|Q|\times|Q|}$ for each symbol $\sigma \in \Sigma$ and vectors $a, b \in \mathbb{N}^{|Q|}$, s.t. $a^T \prod_{j=1}^{|w|} A_{w_j} b = \varphi(f_1(w), \dots, f_m(w))$ for all $w \in \Sigma^\star$. Restricting to words of the form $w = \sigma_1^{n_1}\sigma_2^{n_2}\cdots\sigma_m^{n_m}$ for $n_1, \dots, n_m \in \mathbb{N}$ gives $a^T \left(\prod_{i=1}^m A_{\sigma_i}^{n_i}\right) b = \varphi(n_1, \dots, n_m)$. Using Lemma 3.21 on each of the $A_{\sigma_i}^{n_i}$ we see that every entry of $A_{\sigma_i}^{n_i}$ is an ultimately almost PORC function in n_i. Consequently, every entry of $\prod_{i=1}^m A_{\sigma_i}^{n_i}$ is a finite sum of products of different ultimately almost PORC functions and the same holds for $a^T \left(\prod_{i=1}^m A_{\sigma_i}^{n_i}\right) b = \varphi(n_1, \dots, n_m)$.

We now look at the individual summands of φ and prove that we can rewrite each one as a product of ultimately PORC functions by one-by-one rewriting the exponential constituents. For this let $\varphi^{(1)}(n_1) \cdots \varphi^{(m)}(n_m)$ be one of the summands of φ where $\varphi^{(1)}, \dots, \varphi^{(m)}$ are all ultimately almost PORC functions, with periods $p_1, \dots, p_m$, offsets $N_1, \dots, N_m$ and constituents $\varphi_0^{(i)}, \dots, \varphi_{p_i-1}^{(i)}$ for each $i \in [m]$. If none of the constituents are exponential we are done. Otherwise let $\varphi_j^{(i)}$ be one of the exponential constituents, let $\gamma \in \mathbb{R}^+$ and let $N \in \mathbb{N}$, s.t. $\varphi_j^{(i)}(n_i) \geq 2^{\gamma n_i}$ for $n_i \geq N$. We claim we can set $\varphi_j^{(i)}(n_i) = 0$ without changing the product $\varphi^{(1)}(n_1) \cdots \varphi^{(m)}(n_m)$ for any $n_1, \dots, n_m \in \mathbb{N}$. Call the resulting functions $\psi^{(i)}$ and $\psi_j^{(i)}$. Assume for the sake of contradiction, that there are some $c_1, \dots, c_m \in \mathbb{N}$ where $\varphi^{(1)}(c_1) \cdots \varphi^{(m)}(c_m) \neq \varphi^{(1)}(c_1) \cdots \varphi^{(i-1)}(c_{i-1}) \cdot \psi^{(i)}(c_i) \cdot \varphi^{(i+1)}(c_{i+1}) \cdots \varphi^{(m)}(c_m)$. This implies that $\varphi^{(1)}(c_1) \cdots \varphi^{(i-1)}(c_{i-1}) \cdot \varphi^{(i+1)}(c_{i+1}) \cdots \varphi^{(m)}(c_m) \neq 0$ and $c_i \geq N_i$ as we didn't change any other functions except $\varphi^{(i)}$ for $n_i \geq N_i$.

Constructing constant functions $f'_k(w) = c_k$ for $k \neq i$ and the function $f'_i(w) = p_i \cdot (2^{|x|} + \max(N_i, N)) + j$ which are all in #FA. We see that $\varphi \circ (f'_1, \dots, f'_m) \in$ #FA. Note that for any $w \in \Sigma^\star$ we have $f'_i(w) \geq \max(N_i, N)$ and $f'_i(w) \equiv_{p_i} j$ and thus $\varphi^{(i)} \circ f'_i = \psi^{(i)}_j \circ f'_i$. Combining all of this we again reach a contradiction to the fact that NFAs can only compute at most linearly exponential functions via

$$\begin{aligned}\varphi(f'_1(w), \dots, f'_m(w)) &\geq \varphi^{(1)}(c_1) \cdots \varphi^{(i-1)}(c_{i-1}) \cdot \varphi^{(i)}(f'_i(w)) \cdot \varphi^{(i+1)}(c_{i+1}) \cdots \varphi^{(m)}(c_m) \\ &\geq \varphi^{(i)}(f'_i(w)) \;=\; \varphi^{(i)}_j(f'_i(w)) \;=\; \varphi^{(i)}_j(p_i \cdot (2^{|w|} + \max(N_i, N)) + j) \\ &\geq 2^{\gamma p_i \cdot (2^{|w|} + \max(N_i, N)) + \gamma j}.\end{aligned}$$

Note that the first inequality holds due to all summands of φ being non-negative. ◀

Deciding whether a function φ has such a representation may not always be directly visible, however if φ is a multivariate polynomial we can be more explicit. Every integer-valued multivariate polynomial has integer coefficients when represented in the binomial basis (see [12, Prop. 4.2.1] for a proof of this fact). We say a term $a \cdot \binom{x_1}{d_1} \cdots \binom{x_m}{d_m}$ dominates another term $a' \cdot \binom{x_1}{d'_1} \cdots \binom{x_m}{d'_m}$ if $d_i \geq d'_i$ for all $i \in [m]$. A term is a dominating term of φ if it has non-zero coefficient and it is not dominated by any other term with non-zero coefficient. We can use a similar approach to Lemma 3.16 to rewrite φ as a positive integer linear combination of products of shifted binomials (details can be found in the appendix of the full version).

▶ **Lemma 3.24.** *Let $\varphi(x_1, \dots, x_m) = \sum_{i=1}^{r} a_i \cdot \prod_{j=1}^{m} \binom{x_j}{d_{i,j}}$ with $a_i \in \mathbb{Z}$ and the coefficients of the dominating terms being positive. Then there are $a'_1, \dots, a'_{r'} \in \mathbb{N}$ and $c_1, \dots, c_{r'} \in \mathbb{N}$ with $\varphi(x_1, \dots, x_m) = \sum_{i=1}^{r'} a'_i \cdot \prod_{j=1}^{m} \binom{x_j - c_i}{d_{i,j}}$.*

We can then use a generalization of Lemma 3.17 and replace subtractions $x_i - c'$ by $\max(x_i - c', 0)$. However special care has to be taken for $x_i < c'$, in which case x_i has to be replaced by the corresponding constants first. This adds the additional condition on φ.

▶ **Lemma 3.25.** *Let $\varphi : \mathbb{N}^m \to \mathbb{N}$ be a multivariate polynomial with rational coefficients, such that whenever ψ is formed from φ by replacing any set of variables – including the empty subset – by constants from $\mathbb{N}$, then all dominating terms of ψ have positive coefficients. Then φ is a functional closure property of #FA.*

▶ **Lemma 3.26.** *A multivariate polynomial $\varphi : \mathbb{N}^m \to \mathbb{N}$ with rational coefficients is a functional closure property of #FA iff for every ψ that can be formed from φ by replacing any subset of variables – including the empty set – by constants from $\mathbb{N}$, then all dominating terms of ψ have positive coefficients.*

Proof sketch. Lemma 3.25 already proves that all multivariate polynomials of this form are functional closure properties of #FA. Now let $\varphi : \mathbb{N}^m \to \mathbb{N}$ be a multivariate polynomial with rational coefficient and a functional closure property of #FA. By Theorem 3.23 φ can be written as a finite sum of finite products of ultimately PORC functions. Note that the leading coefficient of each constituent of these ultimately PORC functions is positive. By multivariate polynomial interpolation we can now show that φ is already a finite sum of finite products of these constituents. The dominating terms of φ are then formed by products of the leading coefficients of the constituents and thus are positive. ◀

4 Promise closure properties

▶ **Definition 4.1.** *Let $S \subseteq \mathbb{N}^m$ and let $\varphi : \mathbb{N}^m \to \mathbb{N}$ be a function. We call φ a* functional promise closure property *of #FA with regard to S if for every $f_1, \dots, f_m \in$ #FA defined on some shared alphabet Σ there is a function $g \in$ #FA with $g(w) = \varphi(f_1(w), \dots, f_m(w))$ for every $w \in \Sigma^\star$ for which $(f_1(w), \dots, f_m(w)) \in S$.*

We now want to show that if S fulfils some property, namely admitting polynomial cluster sequences (we postpone the definition to Definition 4.5), then for every functional promise closure property with regard to S there is a functional closure property of #FA that agrees with it on all tuples in S. In other words we can interpolate the functional promise closure property φ on all values of S to obtain a functional closure property φ' for all of #FA. The proof for this follows along the following ideas: First, similar to Theorem 3.23, use the functional closure property φ on the unary counting functions to find an equivalent function φ' that is almost a functional closure property. However after this step some of the constituents of the ultimately almost PORC functions might still be exponential. Theorem 3.23 then proceeded by showing that we can replace all these exponential constituents by the constant zero function, as otherwise we were able to reach a contradiction by constructing functions $f'_1, \dots, f'_m$ and an infinite sequence of inputs $w^{(i)}$, such that $\varphi(f'_1(w^{(i)}), \dots, f'_m(w^{(i)}))$ grows doubly exponential in the length of the inputs. However when dealing with functional promise closure properties we have to be more careful when choosing $f'_1, \dots, f'_m$ and the $w^{(i)}$, because we need $(f'_1(w^{(i)}), \dots, f'_m(w^{(i)})) \in S$ to reach a contradiction. Additionally we don't replace the exponential constituents by the constant zero-function but rather a polynomial that behaves the same for small inputs. For this we need a special variant of univariate polynomial interpolation that yields integer-valued polynomials that are non-negative for all inputs from $\mathbb{N}$:

▶ **Lemma 4.2.** *Let $c_0, \dots, c_N \in \mathbb{N}$. Then there is an integer-valued polynomial $q : \mathbb{Q} \to \mathbb{Q}$ with $q(n) = c_n$ for all $n \in \{0, \dots, N\}$ and $q(n') \geq 0$ for all $n' \in \mathbb{N}$.*

To be able to hit all of S consistently we use independent binary encodings, that allow us to hit all of $\mathbb{N}^m$.

▶ **Lemma 4.3** (Folklore). *The function $f : \{0,1\}^\star \to \mathbb{N}$ defined by being the value of $w \in \{0,1\}^\star$ interpreted as a binary number is in #FA. Additionally it is possible to extend the domain of f to any alphabet $\Sigma \supseteq \{0,1\}$ where the value of f is determined while ignoring any symbols not in $\{0,1\}$.*

For any $n \in \mathbb{N}$ we denote by $\mathrm{bin}(n)$ the unique binary representation of n without leading zeros. We first show the methodology in detail by proving the univariate case.

▶ **Theorem 4.4.** *Let $S \subseteq \mathbb{N}$. Then any function $\varphi : \mathbb{N} \to \mathbb{N}$ is a functional promise closure property of #FA with regard to S iff there is a functional closure property $\psi : \mathbb{N} \to \mathbb{N}$ of #FA with $\varphi_{|S} = \psi_{|S}$.*

Proof. If such a ψ exists, we directly see that φ is a functional promise closure property of #FA with respect to S. Indeed for any $f \in$ #FA we construct $g = \psi \circ f \in$ #FA and see $g(w) = \varphi(f(w))$ for every $w \in \Sigma^\star$ with $f(w) \in S$.

Now on the other hand let φ be any functional promise closure property of #FA with regard to S. Again define the alphabet $\Sigma = \{1\}$ and the function $f : \Sigma^\star \to \mathbb{N}$ with $f(1^n) := n$. Applying the closure property to f gives that there is some $g \in$ #FA with $g(1^n) = \varphi(f(1^n)) = \varphi(n)$ for all $n \in S$. Let $M = (Q, \Sigma, \mathrm{wt}, \mathrm{in}, \mathrm{out})$ be an NFA computing g.

This induces a transition matrix $A \in \mathbb{N}^{|Q|\times|Q|}$ and vectors $a, b \in \mathbb{N}^{|Q|}$, s.t. $a^T A^n b = g(1^n)$ for all $n \in \mathbb{N}$ by Lemma 3.19. We now define $\chi(n) := a^T A^n b$ and see $\chi_{|S} = \varphi_{|S}$. Using Lemma 3.21 on A^n we see that every entry of A^n is an ultimately almost PORC function in n and as such ψ is also an ultimately almost PORC function. Let p be the quasiperiod of χ and let $\chi_0, \ldots, \chi_{p-1}$ be the constituents of χ and let $N \in \mathbb{N}$ be the offset after which χ is defined by the constituents.

We claim that we can now replace every exponential constituent by a polynomial one without changing the value of χ for any $n \in S$. For each $i \in \{0, \ldots, p-1\}$ we distinguish two cases, depending on whether $S_i := S \cap (p\mathbb{Z}+i)$ is finite or it is infinite. If S_i is a finite set, we replace χ_i' with a polynomial that interpolates the same values as χ_i on S_i. Lemma 4.2 ensures that this polynomial is integer-valued and non-negative for all of $\mathbb{N}$. If S_i is an infinite set, we replace χ_i by the constant zero function. Call the resulting ultimately PORC function ψ with constituents ψ_i. Assume for the sake of contradiction there is an $c \in S$, s.t. $\chi(c) \neq \psi(c)$. Clearly such a c would have to be at least N. Now let i be, s.t. $c \in S_i$. It must hold that $\chi_i(c) \neq \psi_i(c)$. Hence S_i cannot be a finite set, since χ_i and ψ_i agree on S_i. Therefore S_i must be infinite. Since χ_i is exponential there is a $\gamma \in \mathbb{R}^+$ and $N' \in \mathbb{N}$, s.t. $\chi_i(n) \geq 2^{\gamma n}$ for all $n \geq N'$. Let $f' \in \#\mathsf{FA}$ be the function of binary evaluation from Lemma 4.3 over the alphabet $\Sigma' = \{0, 1\}$. Then there is a $g' \in \#\mathsf{FA}$ with $g'(\mathrm{bin}(n)) = \varphi(f'(\mathrm{bin}(n))) = \chi(f'(\mathrm{bin}(n)))$ for all $n \in S$. Since S_i is infinite, in particular S_i must contain infinitely many values bigger than $\max(N, N')$. For $n \in S_i$ with $n \geq \max(N, N')$ we now have

$$g'(\mathrm{bin}(n)) = \chi(f'(\mathrm{bin}(n))) = \chi(n) = \chi_i(n) \geq 2^{\gamma n} \geq 2^{\gamma 2^{|\mathrm{bin}(n)|-1}},$$

which is a contradiction to NFAs only being able to only compute functions that are at most linearly exponential in the input length. In conclusion S_i cannot be infinite either and thus c itself cannot exist. ◀

▶ **Definition 4.5.** *A set $S \subseteq \mathbb{N}^m$ admits* polynomial cluster sequences *if for every $N_1, \ldots, N_m \in \mathbb{N}$, and $p_1, \ldots, p_m \in \mathbb{N}$ the projection $\tau : S \to \{0, \ldots, N_1 + p_1 - 1\} \times \ldots \times \{0, \ldots, N_m + p_m - 1\}$ defined by $\tau(n_1, \ldots n_m) = (n_1 \operatorname{srem}_{N_1} p_1, \ldots, n_m \operatorname{srem}_{N_m} p_m)$ has the following property: Any preimage T of a singleton set under τ for every $i \in [m]$ has either bounded i-th coordinate or there is a polynomial $q : \mathbb{N} \to \mathbb{N}$ and an infinite subset $T' \subseteq T$ with unbounded i-th coordinate*[9] *and with $\sum_{j=1}^{m} n_j \leq q(n_i)$ for all $(n_1, \ldots, n_m) \in T'$. We call such an infinite subset a* polynomial cluster sequence *with regards to dimension i.*

We call τ the shifted grid projection *of S with respect to offsets $N_1, \ldots, N_m$ and quasiperiods $p_1, \ldots, p_m$.*

Intuitively this definition requires that each dimension is either bounded or can grow reasonably quickly together with the other dimensions, even when restricted to inputs from specific shifted residue classes. For example $\{(n^2, n^3) \mid n \in \mathbb{N}\}$ admits polynomial cluster sequences, while $\{(n, 2^n) \mid n \in \mathbb{N}\}$ does not.

▶ **Theorem 4.6.** *Let $S \subseteq \mathbb{N}^m$ admit polynomial cluster sequences. Then any function $\varphi : \mathbb{N}^m \to \mathbb{N}$ is a functional promise closure property of $\#\mathsf{FA}$ with regard to S iff there is a functional closure property $\psi : \mathbb{N}^m \to \mathbb{N}$ of $\#\mathsf{FA}$ with $\varphi_{|S} = \psi_{|S}$.*

The technical proof of the theorem can be found in the appendix of the full version. There are multiple natural families for the set S such that S admits polynomial cluster sequences which we describe in the following.

[9] note that this property also follows directly from the polynomial bound on the other coordinates in the i-th coordinate, but we have it as part of the definition for clarity.

▶ **Lemma 4.7.** *Any finite set $S \subseteq \mathbb{N}^m$ admits polynomial cluster sequences.*

Proof. Independent of the offsets and quasiperiods and $i \in [m]$ any subset of S always is finite and thus bounded in every dimension. ◀

An affine variety is defined as the zero set of a finite number of multivariate polynomials. A special case of affine varieties are *graph varieties* (also just called *graphs*, see [20, §2.4, Exe. 12]). An affine variety S is a graph variety if there exist a finite number of j-variate polynomials $\mu_1, \dots, \mu_k$ such that $S = \{(s_1, \dots, s_j, \mu_1(s_1, \dots, s_j), \dots, \mu_k(s_1, \dots, s_j)) \mid (s_1, \dots, s_j) \in \mathbb{Q}^j\} \subseteq \mathbb{Q}^{j+k}$. We call $s_1, \dots, s_j$ the *free* variables, and the remaining variables the *dependent* variables. We call S a monotone graph variety if $\mu_1, \dots, \mu_k$ are all monotone.

▶ **Lemma 4.8.** *Let $S \subseteq \mathbb{Q}^m = \mathbb{Q}^{j+k}$ be a monotone graph variety. Then the set $S \cap \mathbb{N}^m$ admits polynomial cluster sequences.*

The proof of this lemma can be found in the appendix of the full version. All in all this combines to the following theorem which characterizes the special case of multivariate polynomial functional promise closure properties, the technical details can be found in the appendix of the full version.

▶ **Theorem 4.9.** *Let $S \subseteq \mathbb{Q}^m = \mathbb{Q}^{j+k}$ be a monotone graph variety and let $I = I(S)$ be its vanishing ideal. A multivariate polynomial $\varphi : S \to \mathbb{N}$ is a functional promise closure property of #FA with regard to S if and only if there exists a $\psi \in I$ such that $\varphi + \psi$ is a multivariate functional closure property of #FA.*

5 Conclusion

We characterized the functional closure properties of #FA to be precisely the ultimately PORC functions in the univariate case and combinations of ultimately PORC functions in the multivariate case. Additionally we characterize promise functional closure properties of #FA with regard to some natural families of sets S. Natural further directions of research are now whether we can characterize the promise functional closure properties of #FA for more sets S and whether our methods can be applied to characterize functional closure properties for more powerful models of computation.

References

1 Rudolf Ahlswede and David E Daykin. An inequality for the weights of two families of sets, their unions and intersections. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 43:183–185, 1978.
2 Noga Alon and Joel H. Spencer. *The Probabilistic Method, Third Edition*. Wiley-Interscience series in discrete mathematics and optimization. Wiley, 2008.
3 Edwin F Beckenbach and Richard Bellman. *Inequalities*. Springer, Berlin, 1961.
4 Richard Beigel. Closure properties of GapP and #P. In *Proceedings of the Fifth Israeli Symposium on Theory of Computing and Systems*, pages 144–146. IEEE, 1997. doi:10.1109/ISTCS.1997.595166.
5 Swee Hong Chan and Igor Pak. Computational complexity of counting coincidences. *CoRR*, abs/2308.10214, 2023. doi:10.48550/arXiv.2308.10214.
6 Swee Hong Chan and Igor Pak. Equality cases of the alexandrov-fenchel inequality are not in the polynomial hierarchy. *CoRR*, abs/2309.05764, 2023. doi:10.48550/arXiv.2309.05764.
7 David A. Cox, John Little, and Donal O'Shea. *Ideals, varieties, and algorithms – An introduction to computational algebraic geometry and commutative algebra (2. ed.)*. Undergraduate texts in mathematics. Springer, 1997.

8 Manfred Droste, Werner Kuich, and Heiko Vogler. *Handbook of weighted automata*. Springer Science & Business Media, 2009.

9 Manfred Droste and Dietrich Kuske. Weighted automata. In Jean-Éric Pin, editor, *Handbook of Automata Theory*, pages 113–150. European Mathematical Society Publishing House, Zürich, Switzerland, 2021. doi:10.4171/Automata-1/4.

10 HG Hardy, JE Littlewood, and G. Pólya. *Inequalities*. Cambridge University Press, 1952.

11 Ulrich Hertrampf, Heribert Vollmer, and Klaus W Wagner. On the power of number-theoretic operations with respect to counting. In *Proceedings of Structure in Complexity Theory. Tenth Annual IEEE Conference*, pages 299–314. IEEE, 1995. doi:10.1109/SCT.1995.514868.

12 Christian Ikenmeyer and Igor Pak. What is in #P and what is not? In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 860–871. IEEE, 2022. doi:10.1109/FOCS54457.2022.00087.

13 Christian Ikenmeyer, Igor Pak, and Greta Panova. Positivity of the symmetric group characters is as hard as the polynomial time hierarchy. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3573–3586. SIAM, 2023. doi:10.1137/1.9781611977554.ch136.

14 Ines Klimann, Sylvain Lombardy, Jean Mairesse, and Christophe Prieur. Deciding unambiguity and sequentiality from a finitely ambiguous max-plus automaton. *Theoretical Computer Science*, 327(3):349–373, 2004. doi:10.1016/j.tcs.2004.02.049.

15 Igor Pak. Complexity problems in enumerative combinatorics. In *Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018*, pages 3153–3180. World Scientific, 2018.

16 Igor Pak. Combinatorial inequalities. *Notices of the AMS*, 66(7), August 2019.

17 Igor Pak. What is a combinatorial interpretation? to appear: Proc. Open Problems in Algebraic Combinatorics. https://www.samuelfhopkins.com/OPAC/files/proceedings/pak.pdf, 2022.

18 J Peterson. Beviser for wilsons og fermats theoremer. *Tidsskrift for mathematik*, 2:64–65, 1872.

19 Marcel Paul Schützenberger. On the definition of a family of automata. *Inf. Control.*, 4(2-3):245–270, 1961. doi:10.1016/S0019-9958(61)80020-X.

20 Igor R Shafarevich. *Basic algebraic geometry 1: Varieties in projective space*. Springer Science & Business Media, 3 edition, 2013.

21 Michael Spivey. The Chu-Vandermonde identity via Leibniz's identity for derivatives. *The College Mathematics Journal*, 47(3):219–220, 2016.

22 Richard P Stanley. Positivity problems and conjectures in algebraic combinatorics. *Mathematics: frontiers and perspectives*, 295:319, 1999.