Theory Comput Syst (2018) 62:1241–1268
https://doi.org/10.1007/s00224-017-9817-2

The Algebraic Theory of Parikh Automata

Michaël Cadilhac[1] · **Andreas Krebs**[1] ·
Pierre McKenzie[2]

Published online: 7 November 2017

Abstract The Parikh automaton model equips a finite automaton with integer registers and imposes a semilinear constraint on the set of their final settings. Here the theories of typed monoids and of rational series are used to characterize the language classes that arise algebraically. Complexity bounds are derived, such as containment of the unambiguous Parikh automata languages in NC^1. Affine Parikh automata, where each transition applies an affine transformation on the registers, are also considered. Relying on these characterizations, the landscape of relationships and closure properties of the classes at hand is completed, in particular over unary languages.

not the language containment problem!

Extended version of the paper with the same title appearing in the proceedings of the 5th Conference on Algebraic Informatics (CAI'13), LNCS vol. 8080, pp. 60–73, Springer-Verlag (2013). The main changes in contents are as follows: Section 3 contains a new Chomsky-Schützenberger-like characterization of $\mathcal{L}_{\text{CA}}$; Section 4 additionally shows that $\mathcal{L}_{\text{DetAPA}} = \mathcal{L}_{\text{UnAPA}}$; Section 5 presents an expressiveness lemma that allows, in Section 6, to show separation of $\mathcal{L}_{\text{DetAPA}}$ and $\mathcal{L}_{\text{APA}}$, and new nonclosure results; moreover, most proofs were rewritten for added uniformity.

The first two authors are supported by the DFG Emmy Noether program (KR 4042/2). The third author is supported by the Natural Sciences and Engineering Research Council of Canada and by the "Chaire Digiteo, ENS Cachan - École Polytechnique."

✉ Michaël Cadilhac
michael@cadilhac.name

Andreas Krebs
mail@krebs-net.de

Pierre McKenzie
mckenzie@iro.umontreal.ca

1 Wilhelm-Schickard-Institut, Universität Tübingen, Sand 13, 72070 Tübingen, Germany

2 Département d'informatique et de recherche opérationnelle, Université de Montréal, CP 6128 succursale Centre-Ville, Montréal, QC H3C 3J7, Canada

Keywords Automata · Semilinear sets · Algebra · Monoids · Typed monoids · Circuits

1 Introduction

The Parikh automaton model was introduced in [23]. It amounts to a nondeterministic finite automato equipped with registers tallying up the number of occurrences of each transition along an accepting run. Such a run is then deemed successful iff the tuple of final register settings falls within a fixed semilinear set. An *affine* variant of the model in which transitions further induce an affine transformation on the registers was considered in [9]. An *unambiguous* variant of the model was considered in [11]. Expressivity of the model was compared with that of other models, notably with reversal-bounded counter automata, in [9, 23]. The complexity of decision problems and equivalent formulations in terms of expressions were studied in [16]. Tree Parikh automata and other variants were considered in [22]. A model of regular expression equipped with tests on the number of occurrences of letters is investigated in [1, 19].

Klaedtke and Rueß [23] introduced the Parikh automaton as a more general mechanism than the finite automaton, able to capture an extension of weak existential monadic second-order logic with successor that was actually applied in the area of verification. Subsequent variants of the Parikh automaton came about for the purpose of better understanding the model, its language-theoretic expressiveness and the limits beyond which adding features made questions about the model undecidable. Our first motivation in the present paper is to fill some gaps that remain in that understanding. Our contributions in that regard, and partial comparison with previous work, are depicted on Figs. 1 and 2 in Section 7.

Our second motivation is complexity-theoretic. Regular languages are well known to hold the key to much of the internal structure of the complexity class NC^1 of languages computed by families of logarithmic depth boolean circuits (see [29] for a pedagogical account of this fascinating connection). In particular, the complexity classes AC^0, ACC^0 and NC^1 are captured by automata having aperiodic, solvable and nonsolvable so-called syntactic monoids [3, 4] respectively. Another prominent subclass of NC^1, the class TC^0, that initially had no algebraic characterization because of the non-regularity of the MAJ language (of binary words having more ones that zeroes), was captured subsequently in the same algebraic framework by the introduction of typed monoids [25].

In a bit more detail, the 50-year-old algebraic theory of finite automata (see [27]) and the more recent typed monoid framework focus on classes of monoids, called *varieties*, that are closed under natural operations. The core phenomenon there is that distinct monoid varieties *provably* capture distinct classes of languages as inverse homomorphic images of an accepting subset of a monoid [6, 14]. The internal structure of NC^1 would be entirely elucidated if one could at will determine when distinct monoid varieties *still* capture distinct language classes, once the classical notion of a homomorphism is appropriately generalized, i.e., is replaced by the notion of a polynomial length program over the relevant monoid (see [6, 29]).

We show in this paper that variants of the Parikh automaton, which have the ability to count in various more subtle ways than the finite automaton, have elegant algebraic characterizations. As was the case when algebraic automata theory was first linked to circuit complexity, our hope is that developing an algebraic perspective on Parikh automata and their language classes can help towards proving better complexity lower bounds.

Our specific contributions begin with algebraic characterizations of the language classes defined by the deterministic, the unambiguous and the affine variants of the Parikh automaton, where an affine Parikh automaton generalizes the Parikh automaton by allowing each transition to perform an affine transformation on the automaton registers. (We use APA below to refer to the affine Parikh automaton, but CA, for "constrained automaton", to refer to the basic Parikh automaton PA, in deference to the fact that although the subtly different CA and PA are equivalent in terms of expressivity [9], the "automaton with constraint" point of view is closer to the usage we make of the model). Our characterizations are the following (for a brush up on the algebraic terminology please see Section 2):

- The class $\mathcal{L}_{\text{DetCA}}$ of languages accepted by deterministic CA is the set of languages recognized by typed monoids from $\mathbf{Z}^+ \circledcirc \mathbf{M}$, i.e., by wreath products of the monoid of integer vectors with some finite monoid; the smallest typed monoid variety generated by $\mathbf{Z}^+ \circledcirc \mathbf{M}$ also captures $\mathcal{L}_{\text{DetCA}}$;
- The class $\mathcal{L}_{\text{UnCA}}$ of languages accepted by unambiguous CA is the set of languages recognized by typed monoids from $\mathbf{Z}^+ \Box \mathbf{M}$, i.e., by block products of the monoid of integer vectors with some finite monoid; the smallest typed monoid variety generated by $\mathbf{Z}^+ \Box \mathbf{M}$ also captures $\mathcal{L}_{\text{UnCA}}$;
- The classes $\mathcal{L}_{\text{DetAPA}}$ and $\mathcal{L}_{\text{UnAPA}}$, of languages accepted by deterministic and by unambiguous affine Parikh automata respectively are similarly characterized using wreath and block products of the monoid of integer matrices with some finite monoid;
- The class $\mathcal{L}_{\text{DetAPA}}$ is the Boolean closure of the positive supports of rational series over the integers, where the latter are the languages of words with a positive weight in a weighted automaton over $(\mathbb{Z}, +, \times)$.

The characterizations of $\mathcal{L}_{\text{DetCA}}$ and $\mathcal{L}_{\text{UnCA}}$ above add further legitimacy to the theory of typed monoids and further relevance of that theory to our understanding of NC^1. More generally, the algebraic characterizations above allow an almost complete resolution of the expressiveness and closure questions left open in previous work concerning CA and APA. In particular, we draw the following consequences:

- $\mathcal{L}_{\text{UnCA}} \subseteq \text{NC}^1$, a fact which is not immediately obvious from the operation of an unambiguous constrained automaton;
- a Chomsky-Schützenberger-like characterization of $\mathcal{L}_{\text{CA}}$ (and thus of the languages of reversal-bounded counter automata), implying that $\mathcal{L}_{\text{CA}}$ is the closure, under morphisms, inverse morphisms and intersection with a regular language, of the commutative closure of the Dyck languages;
- $\mathcal{L}_{\text{DetAPA}} = \mathcal{L}_{\text{UnAPA}}$;
- the non-closure of $\mathcal{L}_{\text{DetAPA}}$ under concatenation with a regular language, under Kleene closure or under commutation closure;

– the separation of $\mathcal{L}_{\mathrm{APA}}$ and $\mathcal{L}_{\mathrm{DetAPA}}$ by means of a unary language;
– further technical consequences, described once the relevant definitions are introduced and depicted on Figs. 1 and 2 in Section 7.

The structure of this paper is as follows. Section 2 assembles the relevant language-theoretic and automata-theoretic notions, including those of typed monoids and block product. In Section 3, normal forms for CA and APA are developed, with the goal of allowing a uniform treatment of the proofs of the algebraic characterizations to follow, and the Chomsky-Schützenberger-like characterization of $\mathcal{L}_{\mathrm{CA}}$ is given. Section 4 contains the proofs of our algebraic characterizations and justifies the collapse of $\mathcal{L}_{\mathrm{UnAPA}}$ to $\mathcal{L}_{\mathrm{DetAPA}}$. Section 5 relates $\mathcal{L}_{\mathrm{DetAPA}}$ to rational power series. Section 6 draws consequences about membership or non-membership of specific languages in the various classes, locates language classes with respect to each other and deduces new closure or non-closure properties, notably concerning $\mathcal{L}_{\mathrm{DetAPA}}$. Section 7 concludes with a discussion and suggestions for future work.

2 Preliminaries

Monoids, morphisms A monoid is a set M with an associative operation, usually denoted multiplicatively $(x, y) \mapsto xy$, and an identity element denoted 1. For $S \subseteq M$, we write S^* for the monoid generated by S, i.e., the smallest submonoid of M containing S. We write M^{R} for the *reversed* monoid (sometimes called the *opposite* monoid) of M, that is, the monoid with the same base set as M, and the operation reversed: mn in M is equal to nm in M^{R}. We naturally extend this notation to sets of monoids. The powerset of M, written $\mathcal{P}(M)$, is endowed with a monoid structure given by the pointwise multiplication of M.

A (monoid) *morphism* from M to N is a map preserving product and identity. For two monoids M_1 and M_2, we define $\pi_i \colon M_1 \times M_2 \to M_i$, with $i = 1, 2$, as the morphisms which are the projections on the i-th component (i.e., $\pi_1(m_1, m_2) = m_1, \pi_2(m_1, m_2) = m_2$).

Languages The symbols Σ and T (capital tau) will always implicitly refer to some alphabets, i.e., finite sets of symbols. With concatenation as the operation, Σ^*, T^* are monoids where ε, the empty word, is the identity element. Subsets of these monoids are referred to as *languages*. A morphism from Σ^* need only be defined on the elements of Σ. For $w \in \Sigma^*$, we write w^{R} for the *reversal* of $w \in M$, that is, the image of w under the isomorphism from Σ^* to $(\Sigma^*)^{\mathrm{R}}$ which is the identity on Σ. For $L \subseteq \Sigma^*$, we write $\mathsf{Pref}(L)$ for the language $\{u \mid (\exists v)[uv \in L]\}$. We further say that a morphism h from Σ^* is injective (resp. prefix-injective) on L if for any $u, v \in L$ (resp. $u, v \in \mathsf{Pref}(L)$), $h(u) = h(v)$ implies $u = v$. If h further maps to T^*, we say that it is *length-preserving* if $h(\Sigma) \subseteq \mathrm{T}$.

Integers, vectors, matrices We write $\mathbb{Z}, \mathbb{Z}_+$ for the sets of integers and positive integers, respectively. Let $d \in \mathbb{Z}_+$ be some dimension. Vectors in $\mathbb{Z}^d$ are noted in bold, e.g., $\mathbf{v}$ whose elements are $v_1, v_2, \ldots, v_d$. We write $\mathbf{e}_i \in \{0, 1\}^d$ for the vector having a 1 only in position i, and $\mathbf{0}$ for the all-zero vector, where the dimension d is implicit.

We view $\mathbb{Z}^d$ as the additive monoid $(\mathbb{Z}^d, +)$, with $+$ the component-wise addition and $\mathbf{0}$ the identity element. We let $\mathcal{M}_d(\mathbb{Z})$, for $d \geq 1$, be the monoid of square matrices of dimension $d \times d$ with values in $\mathbb{Z}$, under matrix multiplication. We will often speak of the reversal of this monoid, that we simply write $\mathcal{M}_d^{\mathrm{R}}(\mathbb{Z})$. Denoting by M^{tr} the transpose of a matrix M, it thus holds that $(MN)^{\mathrm{tr}}$ (in the monoid $\mathcal{M}_d(\mathbb{Z})$) is equal to the product of M^{tr} and N^{tr} in $\mathcal{M}_d^{\mathrm{R}}(\mathbb{Z})$.

Semilinear sets, Parikh image A subset C of $\mathbb{Z}^d$ is *linear* if there exist $\mathbf{c} \in \mathbb{Z}^d$ and a finite $P \subseteq \mathbb{Z}^d$ such that $C = \mathbf{c} + P^*$. The subset C is said to be *semilinear* if it is equal to a finite union of linear sets: $\{4n + 56 \mid n > 0\}$ is semilinear while $\{2^n \mid n > 0\}$ is not. We will often use the fact that the semilinear sets are the sets of vectors definable in first-order logic with addition [18], i.e., a set $C \subseteq \mathbb{Z}^d$ is semilinear iff there is a first-order formula $\phi(x_1, x_2, \ldots, x_d)$ using addition, order, and constants, such that $\mathbf{x} \in C$ iff $\phi(\mathbf{x})$ holds true. Restricting this view, a *sign set* is a subset of $\mathbb{Z}^d$ that can be expressed as a Boolean combination of conditions of the form $x_i > 0$.

Let $\Sigma = \{a_1, a_2, \ldots, a_n\}$ be an (ordered) alphabet. The *Parikh image* is the morphism $\mathsf{Pkh}\colon \Sigma^* \to \mathbb{Z}^n$ defined by $\mathsf{Pkh}(a_i) = \mathbf{e}_i$, for $1 \leq i \leq n$, with in particular, $\mathsf{Pkh}(\varepsilon) = \mathbf{0}$. For $w \in \Sigma^*$ and $a_i \in \Sigma$, we write $|w|_{a_i}$ for the i-th component of $\mathsf{Pkh}(w)$. The Parikh image of a language L is defined as $\mathsf{Pkh}(L) = \{\mathsf{Pkh}(w) \mid w \in L\}$. The name of this morphism stems from Parikh's theorem [26], stating that for L context-free, $\mathsf{Pkh}(L)$ is semilinear; outside language theory, it is also referred to as the *commutative image*. We call the *commutative closure* of a language L the language $\mathsf{Comm}(L) = \mathsf{Pkh}^{-1}(\mathsf{Pkh}(L))$. An equivalent formulation of Parikh's theorem is that context-free and regular languages have the same commutative closures.

Affine functions A function $f\colon \mathbb{Z}^d \to \mathbb{Z}^d$ is a (total and positive) *affine function* of dimension d if there exist a matrix $M \in \mathcal{M}_d(\mathbb{Z})$ and $\mathbf{v} \in \mathbb{Z}^d$ such that for any $\mathbf{x} \in \mathbb{Z}^d$, $f(\mathbf{x}) = M\mathbf{x} + \mathbf{v}$. We let $\mathcal{F}_d$ be the monoid of such functions under the operation $\diamond$ defined by $(f \diamond g)(\mathbf{x}) = g(f(\mathbf{x}))$, where the identity element is the identity function.

Automata An automaton is a quintuple $A = (Q, \Sigma, \delta, q_0, F)$ where Q is a finite set of states, Σ is an alphabet, $\delta \subseteq Q \times \Sigma \times Q$ is a set of transitions, $q_0 \in Q$ is the initial state, and $F \subseteq Q$ is a set of final states. We view δ as an alphabet, and thus write δ^* for the monoid under concatenation of words over δ. For a transition $t = (q, a, q') \in \delta$, define $\mathsf{From}(t) = q$ and $\mathsf{To}(t) = q'$. We define $\mathsf{Label}_A\colon \delta^* \to \Sigma^*$ as the morphism given by $\mathsf{Label}_A(t) = a$, with, in particular, $\mathsf{Label}_A(\varepsilon) = \varepsilon$, and write Label when A is clear from the context. The set of accepting paths of A, i.e., the set of words over δ describing paths starting from q_0 and ending in F, is written $\mathsf{Run}(A)$. We assume that every state in Q appears along at least one path in $\mathsf{Run}(A)$. The language of the automaton is $L(A) = \mathsf{Label}_A(\mathsf{Run}(A))$. An automaton is *unambiguous* if Label_A is injective on $\mathsf{Run}(A)$, and *deterministic* if Label_A is prefix-injective on $\mathsf{Run}(A)$.

A *constrained automaton (CA)* [9] is a pair (A, C) where A is an automaton with d transitions and $C \subseteq \mathbb{Z}^d$ is semilinear. Its language $L(A, C)$ is the set of labels of accepting paths ρ with $\mathsf{Pkh}(\rho) \in C$, that is, the set $\mathsf{Label}_A(\mathsf{Run}(A) \cap \mathsf{Pkh}^{-1}(C))$. The CA is said to be *deterministic (DetCA)* if A is deterministic, and *unambiguous (UnCA)* if A is unambiguous. We write $\mathcal{L}_{\mathrm{CA}}$, $\mathcal{L}_{\mathrm{DetCA}}$, and $\mathcal{L}_{\mathrm{UnCA}}$ for the classes of languages recognized by CA, DetCA, and UnCA, respectively. Constrained automata are equivalent to a wealth of other computation devices [23], notably Ibarra's reversal-bounded counter machines (written RBCM in Fig. 1) [20], and enjoy a large spectrum of desirable properties. For example, they are closed under intersection, morphisms, inverse morphisms, and commutative closure—it is readily seen from this and the definition of CA that $\mathcal{L}_{\mathrm{CA}}$ is the smallest class containing the regular languages and closed under morphisms, inverse morphisms, intersection, and commutative closure.

An *affine Parikh automaton (APA)* [9] of dimension d is a triple (A, U, C) where A is an automaton with transition set δ, $U : \delta^* \to \mathcal{F}_d$ is a morphism, and $C \subseteq \mathbb{Z}^d$ is semilinear. Its language $L(A, U, C)$ is the set of labels of accepting paths ρ for which $U(\rho)$ maps $\mathbf{0}$ to a value in C, that is, the set $\mathsf{Label}_A(\{\rho \in \mathsf{Run}(A) \mid [U(\rho)](\mathbf{0}) \in C\})$. The APA is said to be *deterministic (DetAPA)* if A is deterministic, and *unambiguous (UnAPA)* if A is unambiguous. We write $\mathcal{L}_{\mathrm{DetAPA}}$ and $\mathcal{L}_{\mathrm{UnAPA}}$ for the classes of languages recognized by DetAPA and UnAPA, respectively.

The rest of this section is concerned with algebraic language theory, with a presentation emphasizing typed monoids [25].

Typed monoids A *typed monoid* [25] is a pair $(S, \mathfrak{S})$ where S is a finitely generated monoid and $\mathfrak{S}$ is a finite Boolean algebra of subsets of S (the *types*, that will later serve as accepting subsets for language recognition). We write this pair succinctly as $S[\mathfrak{S}]$, or simply S if the type set is implicit. If $\mathcal{S} \subseteq S$, then $S[\mathcal{S}]$ is short for $S[\{\emptyset, \mathcal{S}, \overline{\mathcal{S}}, S\}]$. Every finite monoid M is seen as the typed monoid $M[\mathcal{P}(M)]$.

For two typed monoids $M[\mathfrak{M}]$, $N[\mathfrak{N}]$, their direct product $M[\mathfrak{M}] \times N[\mathfrak{N}]$ is the monoid $M \times N$ equipped with the type set which is the Boolean closure of $\{\mathcal{M} \times \mathcal{N} \mid \mathcal{M} \in \mathfrak{M} \text{ and } \mathcal{N} \in \mathfrak{N}\}$.

Recognition A typed monoid $S[\mathfrak{S}]$ *recognizes* a language $L \subseteq \Sigma^*$ if there are a morphism $h : \Sigma^* \to S$ and a type $\mathcal{S} \in \mathfrak{S}$ such that $L = h^{-1}(\mathcal{S})$. We write $\mathcal{L}(S[\mathfrak{S}])$ for the class of languages, over any alphabet, recognized by $S[\mathfrak{S}]$ and extend this notation naturally to classes of typed monoids. Over finite typed monoids, this definition is the same as the traditional one of the theory of Eilenberg [14].

Varieties of languages and monoids A class of languages is said to be a *variety of languages* if it is closed under the Boolean operations, inverse morphisms, and quotient by a word ($u^{-1}L = \{v \mid uv \in L\}$ and its symmetric operation Lu^{-1}). A class of typed monoids is said to be a *variety of typed monoids* if it is closed under division (M divides N if M is a quotient of a submonoid of N) and direct products. We have:

Theorem 1 ([6]) *The languages recognized by the monoids of a variety of monoids form a variety of languages. Conversely, the monoids recognizing the languages of a variety of languages form a variety of monoids. This correspondence is one-to-one.*

The traditional theorem of Eilenberg [14] provides a similar statement for varieties of *finite* monoids and varieties of *regular* languages. In particular, a language is regular iff it is recognized by a finite monoid.

Block and wreath products The algebraic theory of languages is largely based on constructing complicated objects using simple ones and a product operation. In our presentation, we will rely on block and wreath products, and we present the latter as a specialization of the former.

Let $M[\mathfrak{M}]$ be a typed monoid and $N[\mathfrak{N}]$ a finite typed monoid.[1] We describe their block product $M[\mathfrak{M}] \mathbin{\Box} N[\mathfrak{S}]$ in two steps: first the (untyped) monoid and then the type set. The monoid $M \mathbin{\Box} N$ is a subset of $M^{N \times N} \times N$ with the following multiplication:

$$(f, n) \times_{(M \,\Box\, N)} (f', n') = (f(\bullet,\, n'\blacklozenge) +_M f'(\bullet\, n,\, \blacklozenge), \quad nn') \ ,$$

where $\bullet, \blacklozenge$ are understood as placeholders, that is, the right-hand function applied to x, y is $f(x, n'y) +_M f'(xn, y)$. The salient property of this multiplication is better seen on multiple elements; for $(f_i, n_i) \in M \mathbin{\Box} N$, $i \in [k]$, it holds that:

$$\prod_{i \in [k]} (f_i, n_i) = \left(\sum_{i \in [k]} f_i(n_1 n_2 \cdots n_{i-1} \bullet,\, \blacklozenge\, n_{i+1} n_{i+2} \cdots n_k), \quad n_1 n_2 \cdots n_k \right) \ .$$

In words, f_i is applied to the pair consisting of the product of the "past" n_j's, $j < i$, and the "future" n_j's, $j > i$. The type set of $M \mathbin{\Box} N$ is then the Boolean algebra generated by:

$$\{(f, n) \in S \mid f(1, 1) \in \mathcal{M} \wedge n \in \mathcal{N}\} \text{ for each } \mathcal{M} \in \mathfrak{M} \wedge \mathcal{N} \in \mathfrak{N} \ .$$

The wreath product (resp. right wreath product) is the restriction of the block product in which the values of the functions f depend only on their left (resp. right) argument. We write these products $M[\mathfrak{M}] \mathbin{\textcircled{\scriptsize{l}}} N[\mathfrak{N}]$ and $M[\mathfrak{M}] \mathbin{\textcircled{\scriptsize{r}}} N[\mathfrak{N}]$, respectively, and we see these as subsets of $M^N \times N$.

3 Normal Forms of CA and APA

We will frequently focus on languages which do not contain the empty word. This is a technical simplification which introduces no loss of generality, as all our classes of languages at hand will contain $\{\varepsilon\}$ and be closed under union. We provide some normal forms for CA and APA languages that rely only on morphisms and regular languages. Kambites [21, Proposition 2] shows a similar result for $\mathcal{L}_{\mathrm{CA}}$, therein called $\mathbb{Z}^d$-automata.

[1]The restriction that N is finite is needed to preserve the property that the monoids at hand are finitely generated. It is possible to define a sensible product of two infinite typed monoids, see [25]. We will however only need this particular case.

Lemma 1 *Let $L \subseteq \Sigma^+$ be a CA language. There are a length-preserving morphism $h\colon \mathrm{T}^* \to \Sigma^*$, a regular language $R \subseteq \mathrm{T}^*$, and a morphism $g\colon \mathrm{T}^* \to \mathbb{Z}^d$ such that:*

$$L = h(R \cap g^{-1}(\mathbb{Z}_+^d)) \enspace .$$

Moreover, if $L \in \mathcal{L}_{\mathrm{UnCA}}$ (resp. $L \in \mathcal{L}_{\mathrm{DetCA}}$), h can be chosen such that it is injective (resp. prefix-injective) on R.

Proof By definition, the language L of a CA (A, C) is:

$$L = \mathsf{Label}_A(\mathsf{Run}(A) \cap \mathsf{Pkh}^{-1}(C)) \enspace ,$$

and Label_A is length-preserving and injective (resp. prefix-injective) on $\mathsf{Run}(A)$ if A is unambiguous (resp. deterministic). Let us identify Label_A, $\mathsf{Run}(A)$ and Pkh^{-1} with the notations of the statement of the lemma, and thus write:

$$L = h(R \cap g^{-1}(C)) \enspace ,$$

with no hypotheses on h, R, and g other than the ones given in the statement to be proven. Our goal is to turn C into $\mathbb{Z}_+^d$ while preserving these properties.

Recall (e.g., [15]) that for any semilinear set $C \subseteq \mathbb{Z}^d$, there is a Boolean combination of expressions of the form: $\sum_{i \in [d]} \alpha_i x_i > c$ and $\sum_{i \in [d]} \alpha_i x_i \equiv_p c$, with $\alpha_i, c \in \mathbb{Z}$ and $p > 1$, which is true iff $(x_1, x_2, \ldots, x_d) \in C$. Note that the α_i's may be zero.

Let us thus assume that C is expressed as such a Boolean combination in disjunctive normal form. Moreover, the negation of $x \equiv_p c$ being equivalent to $\bigvee_{c' \in [p] \setminus \{c\}} x \equiv_p c'$, and the negation of $\sum \alpha_i x_i > c$ to $\sum (-\alpha_i) x_i > -c - 1$, we may assume that negations do not appear in the formula for C.

Let us now note that expressions of the form of the lemma's statement are closed under union. Indeed, assume h', R', g', d' and h'', R'', g'', d'' verify the lemma for two languages L' and L'', with $h'\colon (\mathrm{T}')^* \to \Sigma^*$ and $h''\colon (\mathrm{T}'')^* \to \Sigma^*$ such that $\mathrm{T}' \cap \mathrm{T}'' = \emptyset$ (we can always ensure this condition). Then:

$$L' \cup L'' = (h' \cup h'')((R' \cup R'') \cap f^{-1}(\mathbb{Z}_+^{d'+d''})) \enspace ,$$

where $f(b) = (g'(b), 1^{d''})$ for $b \in \mathrm{T}'$ and $f(b) = (1^{d'}, g''(b))$ for $b \in \mathrm{T}''$. Moreover, if h' and h'' are injective on R' and R'', respectively, then $h' \cup h''$ is injective on $R' \cup R''$; the same holds for prefix-injectivity, showing the claimed closure property. Now, since we further have that:

$$h(R \cap g^{-1}(C' \cup C'')) = h(R \cap g^{-1}(C')) \cup h(R \cap g^{-1}(C'')) \enspace ,$$

the closure under $\cup$ allows us to assume that C is expressed as a single conjunctive clause.

We first get rid of the $\equiv_p$ atomic formulas. Let $C = C' \cap C''$ where C' is expressed by the conjunction of all the $\equiv$ atomic formulas appearing in C, and C'' the conjunction of the other atomic formulas. Then:

$$h(R \cap g^{-1}(C)) = h(R \cap g^{-1}(C') \cap g^{-1}(C'')) \enspace .$$

Now $R \cap g^{-1}(C')$ is itself a regular language, as an automaton reading w can compute the vector $g(w)$ modulo the different p's appearing as $\equiv_p$ in C', and check that $g(w) \in C'$. Further, if h is (prefix-) injective on R, it is (prefix-) injective on $R \cap g^{-1}(C')$; we thus suppose that C is of the form of C'', that is, expressed as a conjunction of expressions of the form $\sum \alpha_i x_i > c$.

We now show how to replace the constant c appearing in an expression $\sum \alpha_i x_i > c$ with a 0. Let $\dot{\mathrm{T}}$ be a "dotted" version of T, that is, $\dot{\mathrm{T}} = \{\dot{a} \mid a \in \mathrm{T}\}$. Let $R' \subseteq \dot{\mathrm{T}}\mathrm{T}^*$ be the language R where the first letter of each word is replaced with its dotted version; accordingly, let $h' \colon (\mathrm{T} \cup \dot{\mathrm{T}})^* \to \Sigma^*$ be defined by $h'(\dot{a}) = h'(a) = h(a)$, for all $a \in \mathrm{T}$. Further, let $g' \colon (\mathrm{T} \cup \dot{\mathrm{T}})^* \to \mathbb{Z}^{d+1}$ be defined by $g'(a) = (g(a), 0)$ and $g'(\dot{a}) = (g(a), c)$, for all $a \in \mathrm{T}$. Finally, define $C' \subseteq \mathbb{Z}^{d+1}$ as C where the expression under study is rewritten as $\sum_{i \in [d+1]} \alpha_i x_i > 0$ by letting $\alpha_{d+1} = -1$. Now clearly, $h(R \cap g^{-1}(C)) = h'(R' \cap (g')^{-1}(C'))$. Moreover, the property of h being (prefix-) injective on R is carried to h' on R'. The process just presented can be iterated so that C is expressed as a positive conjunction of expressions $\sum \alpha_i x_i > 0$. Let us thus assume this is the case.

As a last step, consider an atomic formula of the form $\sum \alpha_i x_i > 0$; we let g compute the sum in an additional component. Precisely, define $g' \colon \mathrm{T}^* \to \mathbb{Z}^{d+1}$ by $g'(a) = (g(a), \sum_{i \in [d]} \alpha_i (g(a))_i)$, then for a word w, if $(x_1, \ldots, x_d) = g(w)$, then the last component of $g'(w)$ is $\sum_{i \in [d]} \alpha_i x_i$. Thus the atomic formula under study can be replaced by the single test $x_{d+1} > 0$. This process can then be carried out for all such expressions, leading to a conjunction of tests $x_i > 0$. Finally, if i is a dimension that is *not* tested in this conjunction, then the whole dimension can be removed, and C can thus be expressed as $\mathbb{Z}^d$, concluding the proof. □

Writing D_k the Dyck language (i.e., the set of well-parenthesized strings) on k pairs of parentheses, $k > 0$, let us recall Chomsky-Schützenberger's theorem [13]: Any context-free language can be expressed as $h(R \cap D_k)$, where h is a morphism, R a regular language, and $k > 0$. We note, even though we will not make use of this, that similar looking characterizations of $\mathcal{L}_{\mathrm{CA}}$ and related classes can be deduced from Lemma 1. Let us spell out the particular case of $\mathcal{L}_{\mathrm{CA}}$, and slightly strengthen it. Write $D'_k = \mathsf{Comm}(D_k)$, the commutative closure of D_k—it can be easily shown that this is not a context-free language. Then:

Theorem 2 *Any $\mathcal{L}_{\mathrm{CA}}$ language can be expressed as $h(R \cap D'_k)$, for h a morphism and R a regular language. As a consequence, $\mathcal{L}_{\mathrm{CA}}$ is the full trio[2] generated by the languages D'_k.*

Proof Let $L \in \mathcal{L}_{\mathrm{CA}}$; by Lemma 1, $L = h(R \cap g^{-1}(\mathbb{Z}^d_+))$ for R a regular language, $h \colon \mathrm{T}^* \to \Sigma^*$, $g \colon \mathrm{T}^* \to \mathbb{Z}^d$ two morphisms, and some $d > 0$.

Add d fresh letters $a_1, a_2, \ldots, a_d$ to T, and modify h and g so that h erases them and $g(a_i)$ is $-\mathbf{e}_i$. Define $R' = R \cdot a_1^+ a_2^+ \cdots a_d^+$, then clearly $L = h(R' \cap g^{-1}(0^d))$.

Now let P_d be the alphabet of D'_d—we assume the symbols in P_d do not appear in any other alphabet at hand. We order the pairs of parentheses of P_d arbitrarily. Define

[2]A full trio or *cone* is a class of languages closed under morphisms, inverse morphisms, and intersection with the regular languages.

$g' \colon \mathrm{T}^* \to P_d^*$ as follows: if $g(a) = (x_1, x_2, \ldots, x_d)$, then $g'(a)$ is $p_1 p_2 \cdots p_d$ where p_i consists of x_i times the i-th opening parenthesis if $x_i > 0$, and $-x_i$ times the i-th closing parenthesis otherwise, $i \in [d]$. It is readily seen that $L = h(R' \cap (g')^{-1}(D'_d))$.

Let us now see T as a set of *opening* parentheses, and define $\dot{\mathrm{T}}$ as the set of matching closing parentheses. Let $R'' \subseteq (\mathrm{T} \uplus \dot{\mathrm{T}} \uplus P_d)^*$ be the image of R' by the morphism $a \mapsto a\dot{a}g(a)$, where $\dot{a}$ is the matching closing element of a in $\dot{\mathrm{T}}$. Extend h so that it erases all the letters of $\dot{\mathrm{T}}$ and P_d, and let D be the Dyck language on the set of parentheses P_d *and* $(\mathrm{T} \uplus \dot{\mathrm{T}})$. Letting $D' = \mathsf{Comm}(D)$, it holds that $L = h(R'' \cap D')$.

The consequence that $\mathcal{L}_{\mathrm{CA}}$ is the full trio generated by the D'_k is then immediate, as $\mathcal{L}_{\mathrm{CA}}$ is closed under all the trio operations. □

This is, to the best of our knowledge, the first Chomsky-Schützenberger theorem that applies, in particular, to Ibarra's reversal-bounded counter machines [20].

Our normal form for APA is similar, with the notable changes that g maps to matrices, and the expression for the deterministic case is simpler. In the following, we shall consider that a matrix $M \in \mathcal{M}_d(\mathbb{Z})$ is in a subset of $\mathbb{Z}^{d^2}$ if the vector consisting of the concatenation of the *columns* of M is in it.

Lemma 2 *Let $L \subseteq \Sigma^+$ be an APA language. There are a length-preserving morphism $h \colon \mathrm{T}^* \to \Sigma^*$, a regular language $R \subseteq \mathrm{T}^*$, a morphism $g \colon \mathrm{T}^* \to \mathcal{M}_d(\mathbb{Z})$, and a sign set $\mathcal{Z} \subseteq \mathbb{Z}^{d^2}$ such that:*

$$L = h(R \cap g^{-1}(\mathcal{Z})) \enspace .$$

Moreover, if $L \in \mathcal{L}_{\mathrm{UnAPA}}$, then h can be chosen such that it is injective on R. If $L \in \mathcal{L}_{\mathrm{DetAPA}}$, then h and R can be chosen trivial, so that $L = g^{-1}(\mathcal{Z})$ (letting $\mathrm{T} = \Sigma$).

Proof Using [9, Lemma 24], then [9, Lemma 23], we obtain that for any $L \subseteq \Sigma^+$ in $\mathcal{L}_{\mathrm{APA}}$, there is an automaton A with transition set δ, a morphism $g \colon \delta^* \to \mathcal{M}_d^{\mathrm{R}}(\mathbb{Z})$, for some d, a vector $\mathbf{s} \in \mathbb{Z}^d$, and a set C expressed as a Boolean combination of expressions of the form $\sum \alpha_i x_i > c$ such that:

$$L = \mathsf{Label}_A(\mathsf{Run}(A) \cap \{\rho \mid g(\rho)\mathbf{s} \in C\}) \enspace .$$

Moreover, if $L \in \mathcal{L}_{\mathrm{UnAPA}}$, then A is unambiguous (making Label_A injective on $\mathsf{Run}(A)$), and further relying on [9, Remark 35], if $L \in \mathcal{L}_{\mathrm{DetAPA}}$, then:

$$L = \{w \mid g(w)\mathbf{s} \in C\} \enspace .$$

We thus simply need to show that C can be turned into a sign set, while "incorporating" $\mathbf{s}$ into g, so that only the output of g needs to be tested. As a final step, we will modify g so that it works on $\mathcal{M}_d(\mathbb{Z})$ instead of $\mathcal{M}_d^{\mathrm{R}}(\mathbb{Z})$. We do this in the case of DetAPA—these transformations are exactly the same for APA and UnAPA and in case h is injective on $\mathsf{Run}(A)$, this is preserved.

First, we note that the closure under union of expressions of the form $h(R \cap g^{-1}(C))$ still holds, and that negations are not needed to express C. We thus focus on C being a conjunction of expressions of the form $\sum_i \alpha_i x_i > c$.

We may assume that $\mathbf{s}$ contains a coordinate, say j, that is valued 1 and that is preserved by all the matrices $g(w)$—if it is not the case, we can add an extra component to $\mathbf{s}$ and the matrices given by g to obtain just that. As a consequence, $\sum_i \alpha_i x_i > c$ can be written as $\sum_i \alpha_i x_i - c \times x_j > 0$, hence an expression of the form $\sum_i \alpha_i x_i > 0$.

Now, consider one such expression, we extend g to compute the sum. For $a \in \Sigma$, write $g(a) = (R_1, R_2, \cdots, R_d)$, the rows of $g(a)$. Then define $g' \colon \Sigma^* \to \mathcal{M}^{\mathrm{R}}_{(d+1)}(\mathbb{Z})$ by $g'(a) = ((R_1, 0), (R_2, 0), \ldots, (R_d, 0), (\sum_i \alpha_i R_i, 0))$. As a result, if $\mathbf{x} = g(w)\mathbf{s}$ for some word w, then the last component of $g'(w)\mathbf{s}$ is precisely $\sum_i \alpha_i x_i$, and the expression under study can be replaced by $x_{d+1} > 0$; repeating this process shows that C can be assumed to be a sign set.

We then show that the product $g(w)\mathbf{s}$ can be computed within the matrices. For $a \in \Sigma$, write the rows of $g(a)$ as $R_1, R_2, \ldots, R_d$, and define $g' \colon \Sigma^* \to \mathcal{M}^{\mathrm{R}}_{(d+1)}(\mathbb{Z})$ as the morphism mapping $a \in \Sigma$ to the matrix consisting of rows $(R_1, R_1.\mathbf{s})$,$(R_2, R_2.\mathbf{s})$, $\ldots$, $(R_d, R_d.\mathbf{s})$, $\mathbf{0}$. Then for $w \in \Sigma^+$, we have that $g'(w)\mathbf{e}_{d+1} = (g(w)\mathbf{s}, 0)$. Thus checking that $g(w)\mathbf{s} \in C$ is equivalent to checking that the last column of $g'(w)$ is in $C \times \mathbb{Z}$. Hence L can be expressed as $g^{-1}(C)$ for some sign set C.

Finally, to reach the statement of the lemma, we turn g into a morphism mapping to the correct monoid. Define $g' \colon \Sigma^* \to \mathcal{M}_{(d+1)}(\mathbb{Z})$ to be the morphism such that $g'(a) = (g(a))^{\mathrm{tr}}$ for all $a \in \Sigma$. Then, denoting matrix multiplication by a dot, we have inductively that $g'(ua) = g'(u).g'(a) = (g(u))^{\mathrm{tr}}.(g(a))^{\mathrm{tr}} = (g(a).g(u))^{\mathrm{tr}} = (g(ua))^{\mathrm{tr}}$, recalling that in the codomain of g, matrix multiplication is inverted.

Now let $\mathcal{Z}$ be the transpose of C seen as a set of matrices, then $\mathcal{Z}$ is still a sign set, and $L = (g')^{-1}(\mathcal{Z})$. □

4 Algebraic Characterizations of Determinism and Unambiguity

Similar to the untyped algebraic theory of languages, if a typed monoid recognizes a language, it also recognizes its complement. This implies that $\mathcal{L}_{\mathrm{CA}}$, which is not closed under complement, does not admit a typed monoid characterization. We show in this section that deterministic and unambiguous classes do enjoy such a characterization.

4.1 Capturing $\mathcal{L}_{\mathbf{DetCA}}$, $\mathcal{L}_{\mathbf{UnCA}}$, $\mathcal{L}_{\mathbf{DetAPA}}$, and $\mathcal{L}_{\mathbf{UnAPA}}$

Let $\mathbf{M}$ be the variety of typed finite monoids. Let $\mathbf{Z}^+$ be the set of typed monoids $\{\mathbb{Z}[\mathbb{Z}_+]^d \mid d \geq 1\}$. Note that the types of $\mathbb{Z}[\mathbb{Z}_+]^d$ are precisely the sign sets of dimension d.

Theorem 3 $\mathcal{L}(\mathbf{Z}^+ \circledcirc \mathbf{M}) = \mathcal{L}_{\mathrm{DetCA}}$ *and* $\mathcal{L}(\mathbf{Z}^+ \Box \mathbf{M}) = \mathcal{L}_{\mathrm{UnCA}}$.

Proof We show the result for UnCA, the deterministic case following from simple modifications that we present at the end of each direction.

($\mathcal{L}_{\mathrm{UnCA}} \subseteq \mathcal{L}(\mathbf{Z}^+ \Box \mathbf{M})$) Let $L \subseteq \Sigma^*$ be an UnCA language, that is, by Lemma 1:

$$L = h(R \cap g^{-1}(\mathbb{Z}_+^d)) \ ,$$

with $h\colon \mathrm{T}^* \to \Sigma^*$ an injective length-preserving morphism on R, R a regular language, $g\colon \mathrm{T}^* \to \mathbb{Z}^d$ a morphism. When $w \in h(R)$, we see $h^{-1}(w)$ as a single element rather than a singleton.

Let R be recognized by a monoid M, so that $R = \eta^{-1}(E)$ for some morphism η and a subset $E \subseteq M$. We write $[u]$ for $\eta(u)$.

We define a morphism $\phi\colon \Sigma^* \to \mathbb{Z}^d \Box \mathcal{P}(M)$ that recognizes L, by, for $a \in \Sigma$:

$$\begin{aligned} \phi(a) &= (f_a, S_a) \quad \text{where } S_a = \{[b] \mid b \in h^{-1}(a)\} \quad \text{and} \\ f_a(S, S') &= \begin{cases} g(b) & \text{if } \exists! b \in h^{-1}(a), \exists m \in S, m' \in S', m.[b].m' \in E \\ 0^d & \text{otherwise.} \end{cases} \end{aligned} \tag{1}$$

We want to show that if $\phi(w) = (f_w, S_w)$ then (1) $S_w = \{[u] \mid u \in h^{-1}(w)\}$, and (2) if $w \in h(R)$ (that is, S_w consists of a single element and it belongs to E), then $f_w(1, 1) = g(h^{-1}(w))$ (recalling that $h^{-1}(w)$ is a single element by injectivity on R). If this holds, then:

$$L = \phi^{-1}(\mathcal{T}) \text{ where } \mathcal{T} = \{(f, S) \mid f(1, 1) \in \mathbb{Z}_+^d \wedge S \cap E \neq \emptyset\} \ ,$$

which is a type of $\mathbb{Z}[\mathbb{Z}_+]^d \Box \mathcal{P}(M)$, showing the inclusion—note that 1 in the expression $f(1, 1)$ refers to the identity of $\mathcal{P}(M)$, that is, the singleton containing the identity of M.

Point (1) is shown by a simple induction. It holds by definition for $|w| = 1$. Now the second component of $\phi(wa)$ is by induction the product of $S_w = \{[u] \mid u \in h^{-1}(w)\}$ and $S_a = \{[b] \mid b \in h^{-1}(a)\}$. An element of this set is thus of the form $[u][b] = [ub]$ and we indeed have $h(ub) = wa$. Conversely, if $h(ub) = wa$, for some word ub, then $h(u) = w$ and $h(b) = a$ as h is length-preserving, and thus $[u] \in S_w$ and $[b] \in S_a$, proving (1).

We show (2). Suppose $w \in h(R)$, and write $w = a_1 a_2 \cdots a_n$ and $h^{-1}(w) = b_1 b_2 \cdots b_n$. For all $i \in [n]$, we show that $f_{a_i}(S_{a_1 a_2 \cdots a_{i-1}}, S_{a_{i+1} a_{i+2} \cdots a_n}) = g(b_i)$. This implies (2) by the following chain of equalities:

$$\begin{aligned} f_w(1, 1) &= f_{a_1}(1, S_{a_2 a_3 \cdots a_n}) + f_{a_2}(S_{a_1}, S_{a_3 a_4 \cdots a_n}) + \cdots + f_{a_n}(S_{a_1 a_2 \cdots a_{n-1}}, 1) \\ &= g(b_1) + g(b_2) + \cdots + g(b_n) \\ &= g(b_1 b_2 \cdots b_n) = g(h^{-1}(w)) \ . \end{aligned}$$

Thus let $i \in [n]$, and write $u = a_1 a_2 \cdots a_{i-1}$, $u' = a_{i+1} a_{i+2} \cdots a_n$ and v, v' their respective counterparts in $h^{-1}(w)$, that is $v = b_1 b_2 \cdots b_{i-1}$ and $v' = b_{i+1} b_{i+2} \cdots b_n$. Since $u a_i u' \in h(R)$, $[h^{-1}(u a_i u')] \in E$, an element of the set $[v].S_{a_i}.[v']$. Now $[v] \in S_u$ and $[v'] \in S_{u'}$, thus it holds that there is a $b \in h^{-1}(a_i)$ and $m \in S_u$, $m' \in S_{u'}$ such that $m.[b].m' \in E$—b_i is such a b. Suppose now that there is another such b, say b', for other values of m, m', say μ, μ'. As $\mu \in S_u$, by (1) there is an $s \in \mathrm{T}^*$ such that $h(s) = u$ and $\mu = [s]$; a similar statement holds for an s' with respect to μ'. Thus on the one hand $[v].[b].[v'] \in E$ (implying $vbv' \in R$) and $h(vbv') = w$, and on the other hand $[s].[b'].[s'] \in E$ (implying $sb's' \in R$) and $h(sb's') = w$. By injectivity, $vbv' = sb's'$, and as h is length preserving, $s = v$, $b = b'$, and $v' = s'$, showing that the above value of b is unique, and thus that $f_{a_i}(S_u, S_{u'}) = g(b_i)$.

(Modifications for $\mathcal{L}_{\mathrm{DetCA}} \subseteq \mathcal{L}(\mathbf{Z}^+ \circledcirc \mathbf{M})$) If L is a DetCA language, then, by Lemma 1, h is prefix-injective on R. Moreover, it always holds that $w \in \Sigma^*$ is such that $\pi_2(\phi(w)) \cap E \neq \emptyset$ if and only if $w \in h(R)$. Now consider $w = uau' \in h(R)$ with $a \in \Sigma$ and let v be the only element in $h^{-1}(u)$ (by prefix-injectivity). By prefix-injectivity again, there is a unique b such that $b \in h^{-1}(a)$ and vb can be extended on the right to a word in R. This means that there is a unique $b \in h^{-1}(a)$ such that there is a $m \in S_u$ (which in fact consists of the single element $[v]$) and a $m' \in M$ such that $m.[b].m' \in E$. This shows that the condition of Eq. 1 can be replaced by: "if $\exists! b \in h^{-1}(a), \exists m \in S$, $\exists m' \in M, m.[b].m' \in E$," hence the functions f_a can be defined so as not to depend on their second argument. This in turn implies that $L \in \mathcal{L}(\mathbf{Z}^+ \circledcirc \mathbf{M})$.

($\mathcal{L}(\mathbf{Z}^+ \,\square\, \mathbf{M}) \subseteq \mathcal{L}_{\mathrm{UnCA}}$) Let $L \subseteq \Sigma^*$ be recognized by $\mathbb{Z}[\mathbb{Z}_+]^d \,\square\, M$ using a type $\mathcal{T}$ and a morphism $h\colon \Sigma^* \to \mathbb{Z}^d \,\square\, M$, and write for convenience $h(w) = (f_w, m_w)$. As $\mathcal{L}_{\mathrm{UnCA}}$ is closed under the Boolean operations, we may assume that the type $\mathcal{T}$ is of the form:

$$\mathcal{T} = \{(f, m) \mid f(1,1) \in \mathcal{Z} \wedge m \in E\} \ ,$$

for some sign set $\mathcal{Z}$ and $E \subseteq M$.

For any $(s_1, s_2) \in M \times M$, let $A(s_1, s_2)$ be the automaton defined as the quintuple $(M \times M, \Sigma, \delta, (s_1, s_2), M \times \{1\})$ where:

$$\begin{aligned}\delta = \{&((m_1, m_2) - a - (m_1', m_2')) \mid \\ &m_1' = m_1 m_a \text{ and } m_a m_2' = m_2 \in M \text{ and } a \in \Sigma\} \ .\end{aligned}$$

Note that $w \in L(A(s_1, s_2))$ implies $m_w = s_2$. We argue that $A(s_1, s_2)$ is unambiguous for any $(s_1, s_2) \in M \times M$. We show that for any $w \in \Sigma^*$ and any $(s_1, s_2) \in M \times M$, w is the label of at most one accepting path in $A(s_1, s_2)$, by induction on $|w|$. If $w = \varepsilon$, then every $A(s_1, s_2)$ has at most one accepting path labeled w. Now let $w = a \cdot v$ for $v \in \Sigma^*$. Suppose $w \in L(A(s_1, s_2))$. This implies that $m_w = s_2$. The states that can be reached from (s_1, m_w) reading a are all of the form $(s_1 m_a, m)$, $m \in M$. Now v should be accepted by the automaton A where the initial state is set to one of these states; thus there is only one state fitting, $(s_1 m_a, m_v)$. By induction hypothesis, there is only one path in $A(s_1 m_a, m_v)$ recognizing v, thus there is only one path in $A(s_1, m_w)$ recognizing w. This shows that for any s_1, s_2, $A(s_1, s_2)$ is unambiguous.

Let C be the semilinear set consisting of the elements:

$$(x_1, x_2, \ldots, x_{|\delta|}) \text{ s.t. } \sum_{i \in [|\delta|]} x_i \times f_{\mathsf{Label}(t_i)}(\pi_1(\mathsf{From}(t_i)), \pi_2(\mathsf{To}(t_i))) \in \mathcal{Z} \ . \tag{2}$$

We show that $\bigcup_{m \in E} L(A(1, m), C)$ is L, concluding the proof as $\mathcal{L}_{\mathrm{UnCA}}$ is closed under union. Let $w = w_1 w_2 \cdots w_n \in \Sigma^*$. There is a unique accepting path in

$A(1, m_w)$ (and in no other $A(1, m)$) labeled w, and it is going successively through the states $(1, m_w) = (m_\varepsilon, m_w)$, $(m_{w_1}, m_{w_2 w_3 \cdots w_n})$, …, $(m_w, m_\varepsilon) = (m_w, 1)$.. For this path, the sum computed by the semilinear set is:

$$\sum_{i \in [n]} f_{w_i}(m_{w_1 \cdots w_{i-1}}, m_{w_{i+1} \cdots w_n}) \ .$$

This is precisely $f_w(1, 1)$, and checking whether it is in $\mathcal{Z}$ amounts to checking whether $h(w) \in \mathcal{T}$, thus $L = \bigcup_{m \in E} L(A(1, m), C)$.

($\text{Modifications for } \mathcal{L}(\mathbf{Z}^+ \circledcirc \mathbf{M}) \subseteq \mathcal{L}_{\text{DetCA}})$) First note that the automaton constructed is deterministic when we consider its state set to be only the first copy of M. Now if $L \in \mathcal{L}(\mathbf{Z}^+ \circledcirc \mathbf{M})$, then the elements of C from Eq. 2 are definable with only the first arguments of each f. This means that there is no need to keep the second component of the state set, hence the resulting automaton is deterministic. $\square$

Let $\mathbf{ZMat}^+$ be the set of typed monoids $\mathcal{M}_d(\mathbb{Z})$ for any d with the sign sets of dimension $d \times d$ as types.

Theorem 4 $\mathcal{L}(\mathbf{ZMat}^+) = \mathcal{L}_{\text{DetAPA}}$.

Proof ($\mathcal{L}_{\text{DetAPA}} \subseteq \mathcal{L}(\mathbf{ZMat}^+)$) This is a direct consequence of Lemma 2.

($\mathcal{L}(\mathbf{ZMat}^+) \subseteq \mathcal{L}_{\text{DetAPA}}$) The inclusion $\mathcal{L}(\mathbf{ZMat}^+) \subseteq \mathcal{L}((\mathbf{ZMat}^+)^{\text{R}})$ is straightforward. Indeed, similarly to the proof of Lemma 2, we can rely on matrix transposition to simulate the former by the latter.

Now, let $L \in \mathcal{L}((\mathbf{ZMat}^+)^{\text{R}})$, that is, $L = h^{-1}(\mathcal{Z})$, for $h\colon \Sigma^* \to \mathcal{M}_d^{\text{R}}(\mathbb{Z})$, $d \geq 1$, and a sign set $\mathcal{Z}$ of dimension $d \times d$; we show that $L \in \mathcal{L}_{\text{DetAPA}}$. As $\mathcal{L}_{\text{DetAPA}}$ is closed under the Boolean operations, we may assume that $\mathcal{Z}$ is expressed by a single expression $x_{i,j} > 0$.

For any word w, we have that $h(w) \in \mathcal{Z}$ iff $h(w)\mathbf{e}_j \in C$ where C is expressed as $x_i > 0$.

Now let $A = (\{r, s\}, \Sigma, \delta, r, \{s\})$, with $\delta = \{r, s\} \times \Sigma \times \{s\}$. Then let $U : \delta^* \to \mathcal{F}_d$ for $q \in \{r, s\}$, $a \in \Sigma$, and $\mathbf{x} \in \mathbb{Z}^d$ be defined by:

$$[U((q - a - s))](\mathbf{x}) = \begin{cases} h'(a)\mathbf{e}_j & \text{if } q = r, \\ h'(a)\mathbf{x} & \text{otherwise.} \end{cases}$$

This implies that for $w \in \Sigma^+$ and ρ its unique accepting path in A, it holds that $[U(\rho)](\mathbf{0})$ is $h(w)\mathbf{e}_j$. Thus $L(A, U, C) = h^{-1}(\mathcal{Z})$. $\square$

A proof mimicking the construction of Theorem 3 directly shows that $\mathcal{L}(\mathbf{ZMat}^+ \circledcirc \mathbf{M}) \subseteq \mathcal{L}_{\text{DetAPA}}$, hence Theorem 4 implies:

Corollary 1 $\mathcal{L}(\mathbf{ZMat}^+ \circledcirc \mathbf{M}) = \mathcal{L}(\mathbf{ZMat}^+)$.

Theorem 5 $\mathcal{L}(\mathbf{ZMat}^+ \,\square\, \mathbf{M}) = \mathcal{L}_{\text{UnAPA}}$.

Proof $\mathcal{L}_{\text{UnAPA}} \subseteq \mathcal{L}(\mathbf{ZMat}^+ \,\square\, \mathbf{M})$ is the same as $\mathcal{L}_{\text{UnCA}} \subseteq \mathcal{L}(\mathbf{Z}^+ \,\square\, \mathbf{M})$ in Theorem 3, thanks to Lemma 2.

$\mathcal{L}(\mathbf{ZMat}^+ \square \mathbf{M}) \subseteq \mathcal{L}_{\mathrm{UnAPA}}$ is the same as $\mathcal{L}(\mathbf{Z}^+ \square \mathbf{M}) \subseteq \mathcal{L}_{\mathrm{UnCA}}$ in Theorem 3 for the automaton part, and the same as Theorem 4 for the constraint set and affine function parts. □

Remark 1 The properties of Lemmata 1 and 2 thus characterize the related classes. For instance, with the notations of Lemma 1, any language expressible as $h(R \cap g^{-1}(\mathbb{Z}_+^d))$ with h injective on R belongs to $\mathcal{L}_{\mathrm{UnCA}}$.

4.2 $\mathcal{L}_{\mathrm{UnAPA}}$ Collapses to $\mathcal{L}_{\mathrm{DetAPA}}$

So as not to lead the reader into thinking that we need to keep treating $\mathcal{L}_{\mathrm{DetAPA}}$ and $\mathcal{L}_{\mathrm{UnAPA}}$ separately, we provide a proof that these two classes coincide here, without delaying it to Section 6, dedicated to the consequences of the algebraic characterizations. Although an automata-theoretic proof of $\mathcal{L}_{\mathrm{DetAPA}} = \mathcal{L}_{\mathrm{UnAPA}}$ is possible (with arguments similar to [10, Lemma 5]), we provide a purely algebraic proof that relies on sensibly different ideas. This sheds a different light on the reasons why unambiguity does not always provide more expressiveness.

Beyond the characterizations of $\mathcal{L}_{\mathrm{DetAPA}}$ (Theorem 4) and $\mathcal{L}_{\mathrm{UnAPA}}$ (Theorem 5), proving the collapse of $\mathcal{L}_{\mathrm{UnAPA}}$ to $\mathcal{L}_{\mathrm{DetAPA}}$ involves two technical steps. The first (Lemma 3) shows that a block product can be decomposed into a wreath product and a right wreath product. The second shows the closure under reversal of $\mathcal{L}_{\mathrm{DetAPA}}$ and implies (Lemma 5) that Corollary 1 could have been stated with ⓡ rather than ⓦ. Taking stock of both lemmas, the collapse of $\mathcal{L}_{\mathrm{UnAPA}}$ to $\mathcal{L}_{\mathrm{DetAPA}}$, stated as Theorem 6, follows:

Theorem 6 $\mathcal{L}_{\mathrm{UnAPA}} = \mathcal{L}_{\mathrm{DetAPA}}$.

Proof This follows from the following chain:

$$\begin{aligned}
\mathcal{L}_{\mathrm{UnAPA}} &= \mathcal{L}(\mathbf{ZMat}^+ \square \mathbf{M}) && \text{(By Theorem 5)}\\
&\subseteq \mathcal{L}((\mathbf{ZMat}^+ \circledR \mathbf{M}) \circledw \mathbf{M}) && \text{(By Lemma 3)}\\
&= \mathcal{L}((\mathbf{ZMat}^+ \circledw \mathbf{M}) \circledw \mathbf{M}) && \text{(By Lemma 5)}\\
&= \mathcal{L}(\mathbf{ZMat}^+) && \text{(By Corollary 1 twice)}\\
&= \mathcal{L}_{\mathrm{DetAPA}} && \text{(By Theorem 4.)}
\end{aligned}$$
□

Lemma 3 *Let M be a typed monoid and N a finite typed monoid. Then:*

$$\mathcal{L}(M \square N) \subseteq \mathcal{L}((M \circledR N) \circledw N) \enspace .$$

Proof Let $h\colon \Sigma^* \to M \square N$ be a morphism. We define a morphism $h'\colon \Sigma^* \to (M \circledR N) \circledw N$ closely mimicking h as follows. For $a \in \Sigma$, write $h(a) = (f_a, n_a)$. Then $h'(a) = (f'_a, n_a)$, where for $n \in N$, $n \in N$, $f'_a(n) \in M \circledR N$ is defined by:

$$f'_a(n) = (f'_{a,n}, n_a) \quad \text{with } f'_{a,n}(n') = f_a(n, n') \enspace .$$

We now show that if $h(w) = (f_w, n_w)$, then $h'(w) = (f'_w, n_w)$ where f'_w is such that $\pi_1(f'_w(n))(n') = f_w(n, n')$ for any $n, n' \in N$; the part on n_w is already clear.

Suppose $w = a \in \Sigma$, then we have $\pi_1(f'_a(n)) = f'_{a,n}$, thus by definition, $\pi_1(f'_a(n))(n') = f_a(n, n')$.

Suppose now that $w = ua$, for $u \in \Sigma^+$ and $a \in \Sigma$. Then $f_w = f_u(\bullet, n_a\blacklozenge) + f_a(\bullet n_u, \blacklozenge)$. Now $h'(ua) = h'(u)h'(a)$, hence $f'_w = f'_u(\bullet) +_{(M ⓡ N)} f'_a(\bullet n_u)$, and:

$$\begin{aligned} f'_w(n) &= f'_u(n) +_{(M ⓡ N)} f'_a(nn_u) \\ &= (f'_{u,n}, n_u) +_{(M ⓡ N)} (f'_{a,nn_u}, n_a) \\ &= (f'_{u,n}(n_a\bullet) + f'_{a,nn_u}, n_u n_a) \ . \end{aligned}$$

The induction hypothesis implies that $f'_{u,n}(n') = f_u(n, n')$, hence:

$$\pi_1(f'_w(n))(n') = f_u(n, n_a n') + f_a(nn_u, n') = f_w(n, n') \ .$$

Now let $\mathcal{T}$ be a type of $M \,\square\, N$ expressed as:

$$\mathcal{T} = \{(f, n) \mid f(1, 1) \in \mathcal{M} \wedge n \in \mathcal{N}\} \ .$$

Then $\mathcal{T}_1 = \{(f, n) \mid f(1) \in \mathcal{M} \wedge n \in \mathcal{N}\}$ is a type of $M ⓡ N$, and in turn, $\mathcal{T}_2 = \{(f, n) \mid f(1) \in \mathcal{T}_1\}$ is a type of $(M ⓡ N) ⓛ N$. We then have that $h^{-1}(\mathcal{T}) = h'^{-1}(\mathcal{T}_2)$, concluding the proof. (If $\mathcal{T}$ is a Boolean combination of types, then we will obtain an equivalent Boolean combination of types of $(M ⓡ N) ⓛ N$.) □

Proposition 1 $\mathcal{L}_{\text{DetAPA}}$ *is closed under reversal.*

Proof Let $L \in \mathcal{L}(\mathbf{ZMat}^+)$, there are $h\colon \Sigma^* \to \mathcal{M}_d(\mathbb{Z})$ and a sign set $\mathcal{Z} \subseteq \mathbb{Z}^{d^2}$ such that $L = h^{-1}(\mathcal{Z})$. Define $h'\colon \Sigma^* \to \mathcal{M}_d(\mathbb{Z})$ by $h'(a) = (h(a))^{\text{tr}}$. Then for a word w, $h(w) = (h'(w^{\text{R}}))^{\text{tr}}$, and thus $h'(w) \in \mathcal{Z}^{\text{tr}}$ iff $h(w^{\text{R}}) \in \mathcal{Z}$, where we naturally extend the transpose notation to set of matrices. Hence the reversal of L is $(h')^{-1}(\mathcal{Z}^{\text{tr}}) \in \mathcal{L}(\mathbf{ZMat}^+)$. □

Lemma 4 *Let M, N be typed monoids. For any language L, $L \in \mathcal{L}(M ⓛ N)$ iff $L^{\text{R}} \in \mathcal{L}(M^{\text{R}} ⓡ N^{\text{R}})$.*

Proof Let $L \in \mathcal{L}(M ⓛ N)$ be recognized by a morphism $h\colon \Sigma^* \to M ⓛ N$ and a (type) set E, that is, $L = h^{-1}(E)$. Let h' be the morphism from Σ^* to $M^{\text{R}}M^{\text{R}} ⓡ N^{\text{R}}N^{\text{R}}$ that agrees with h on Σ. Write $+$ for the operation of M and $\times$ for the operation of N, and $+^{\text{R}}$ and $\times^{\text{R}}$ for their reversed versions, respectively.

We show that for all words $w \in \Sigma^*$, $h(w) = h'(w^{\text{R}})$. If $|w| = 1$, this is immediate. Let $w = ua$ with $a \in \Sigma$, and write $h(u) = (f_u, n_u)$ and $h(a) = h'(a) = (f_a, n_a)$. We first deal with the first component. For $h(ua)$, it is $f_u(\bullet) + f_a(\bullet \times n_u)$. For $h'((ua)^{\text{R}})$, it is $f_a(n_u \times^{\text{R}} \bullet) +^{\text{R}} f_u(\bullet)$ by the induction hypothesis, that is, $f_u(\bullet) + f_a(\bullet \times n_u)$. We now consider the second component. For $h(ua)$, it is $n_u \times n_a$ where n_a is the second component of $h(a) = h'(a)$. For $h'((ua)^{\mathrm{R}})$, it is $n_a \times^{\mathrm{R}} n_u$ by the induction hypothesis, that is, $n_u \times n_a$, proving the claim.

Hence $w \in h^{-1}(E)$ iff $w^{\mathrm{R}} \in h'^{-1}(E)$, thus $L^{\mathrm{R}} \in \mathcal{L}(M^{\mathrm{R}} \circledR N^{\mathrm{R}})$. □

As $\mathcal{L}_{\mathrm{DetAPA}} = \mathcal{L}(\mathbf{ZMat}^{+} \circledl \mathbf{M})$, this goes on to show:

Lemma 5 $\mathcal{L}(\mathbf{ZMat}^{+} \circledl \mathbf{M}) = \mathcal{L}(\mathbf{ZMat}^{+} \circledR \mathbf{M})$.

Proof Let $L \in \mathcal{L}(\mathbf{ZMat}^{+} \circledl \mathbf{M})$. By Proposition 1, L^{R} is also in $\mathcal{L}(\mathbf{ZMat}^{+} \circledl \mathbf{M})$. Lemma 4 then implies that $(L^{\mathrm{R}})^{\mathrm{R}} = L \in \mathcal{L}((\mathbf{ZMat}^{+})^{\mathrm{R}} \circledR \mathbf{M}^{\mathrm{R}})$, where clearly $\mathbf{M}^{\mathrm{R}} = \mathbf{M}$. Now write the usual matrix implicitly and $\cdot^{\mathrm{R}}$ for its reversed version, then $M_1 \cdot^{\mathrm{R}} M_2 = ({M_1}^{\mathrm{tr}} {M_2}^{\mathrm{tr}})^{\mathrm{tr}}$. Let $h\colon \Sigma^* \to \mathcal{M}_d^{\mathrm{R}}(\mathbb{Z}) \circledR N$ be a morphism for some finite monoid N. We define $h'\colon \Sigma^* \to \mathcal{M}_d(\mathbb{Z}) \circledR N$ recognizing the same language. For this, it is enough to transpose all matrices appearing in the definition of h, that is, with $h(a) = (f_a, n_a)$:

$$\forall a \in \Sigma, \quad h'(a) = (f'_a, n_a) \quad \text{where } f'_a = (f_a(\bullet))^{\mathrm{tr}} \ .$$

We show that the above property of f'_a extends to words by induction, that is, writing $h(w) = (f_w, n_w)$ for any word w, it holds that $h'(w) = (f'_w, n_w)$ where $f'_w = (f_w(\bullet))^{\mathrm{tr}}$.

Let $w = ua$, $u \in \Sigma^{+}, a \in \Sigma$. Then $f_w = f_u \cdot n_a \cdot^{\mathrm{R}} f_a$, thus for any $n \in N$, $f_w(n) = f_u(n_a n) \cdot^{\mathrm{R}} f_a(n)$. Now by induction hypothesis, $f'_w(n) = (f_u(n_a n))^{\mathrm{tr}}(f_a(n))^{\mathrm{tr}}$, implying that $f'_w(n) = (f_u(n_a n) \cdot^{\mathrm{R}} f_a(n))^{\mathrm{tr}} = (f_w(n))^{\mathrm{tr}}$.

Now if $L = h^{-1}(\mathcal{T})$ with $\mathcal{T}$ a type, then:

$$L = h'^{-1}(\mathcal{T}') \quad \text{where } \mathcal{T}' = \{(f', n) \mid (f, n) \in \mathcal{T} \text{ with } f(n) = (f'(n))^{\mathrm{tr}}\} \ ,$$

which clearly is a type of $\mathbf{ZMat}^{+} \circledR \mathbf{M}$, showing that $L \in \mathcal{L}(\mathbf{ZMat}^{+} \circledR \mathbf{M})$. The converse direction is similar. □

5 An Alternative Characterization of $\mathcal{L}_{\mathbf{DetAPA}}$

In this section, we show that the languages of DetAPA are those expressible as a Boolean combination of positive supports of $\mathbb{Z}$-valued rational series, i.e., of sets of words w having the property that, once matrices assigned to each letter in w are multiplied out to form a matrix M, a fixed linear combination of the entries in M yields a positive integer. Such a characterization of $\mathcal{L}_{\mathrm{DetAPA}}$ largely owes to Lemma 2 where, intuitively, Boolean closures were exploited in order to entrust all negations and linear combinations to the matrix multiplications performed in the course of a DetAPA computation. With this characterization of $\mathcal{L}_{\mathrm{DetAPA}}$ in hand, we state in closing this section a bound on the prefix diversity (defined below) of languages in $\mathcal{L}_{\mathrm{DetAPA}}$, setting the stage for the nonclosure and expressiveness properties to be proven in Section 6.2.

Definition 1 (e.g., [7]) Functions from Σ^* into $\mathbb{Z}$ are called ($\mathbb{Z}$-)*series*. For such a series r, it is customary to write (r, w) for $r(w)$. We write $\mathsf{supp}_+(r)$ for the *positive support* of r, i.e., $\{w \mid (r, w) > 0\}$.

A *linear representation* of dimension $d \geq 1$ is a triple $(\mathbf{s}, h, \mathbf{g})$ such that $\mathbf{s} \in \mathbb{Z}^d$ is a *row* vector, $\mathbf{g} \in \mathbb{Z}^d$ is a *column* vector, and $h\colon \Sigma^* \to \mathcal{M}_d(\mathbb{Z})$ is a morphism. It defines the series $r = ||(\mathbf{s}, h, \mathbf{g})||$ with $(r, w) = \mathbf{s}h(w)\mathbf{g}$.

A series is said to be *rational* if it is defined by a linear representation. We write $\mathbb{Z}^{\mathrm{rat}}\langle\!\langle\Sigma^*\rangle\!\rangle$ for the set of rational series.

For a class $\mathcal{C}$ of languages, write $\mathsf{BC}(\mathcal{C})$ for the Boolean closure of $\mathcal{C}$. We have:

Theorem 7 *Over any alphabet* Σ, $\mathcal{L}_{\mathrm{DetAPA}} = \mathsf{BC}(\mathsf{supp}_+(\mathbb{Z}^{\mathrm{rat}}\langle\!\langle\Sigma^*\rangle\!\rangle))$.

Proof ($\mathcal{L}_{\mathrm{DetAPA}} \subseteq \mathsf{BC}(\mathsf{supp}_+(\mathbb{Z}^{\mathrm{rat}}\langle\!\langle\Sigma^*\rangle\!\rangle))$) First note that there is a rational series r such that $\mathsf{supp}_+(r) = \{\varepsilon\}$. Let L be in $\mathcal{L}_{\mathrm{DetAPA}}$; we may thus suppose that $\varepsilon \notin L$. By Lemma 2, let $h\colon \Sigma^* \to \mathcal{M}_d(\mathbb{Z})$ and $\mathcal{Z}$ a sign set such that $L = h^{-1}(\mathcal{Z})$. We wish to show that $L \in \mathsf{BC}(\mathsf{supp}_+(\mathbb{Z}^{\mathrm{rat}}\langle\!\langle\Sigma^*\rangle\!\rangle))$, thus we can assume that $\mathcal{Z}$ is expressed as a single expression $x_{i,j} > 0$. Hence the triple $(\mathbf{e}_i, h, \mathbf{e}_j)$ is a linear representation of a rational series r which associates w to the (i, j) entry of $h(w)$, and thus $L = \mathsf{supp}_+(r)$.

($\mathsf{BC}(\mathsf{supp}_+(\mathbb{Z}^{\mathrm{rat}}\langle\!\langle\Sigma^*\rangle\!\rangle)) \subseteq \mathcal{L}_{\mathrm{DetAPA}}$) As $\mathcal{L}_{\mathrm{DetAPA}}$ is closed under all the Boolean operations, we need only show that $\mathsf{supp}_+(\mathbb{Z}^{\mathrm{rat}}\langle\!\langle\Sigma^*\rangle\!\rangle) \subseteq \mathcal{L}_{\mathrm{DetAPA}}$. Let $(\mathbf{s}, h, \mathbf{g})$ be a linear representation of dimension d of a rational series r over the alphabet Σ. As in Lemma 2, we can embed the computation of $\mathbf{s}h(w)\mathbf{g}$ into h. More precisely, we described in the proof thereof how we can build $h'\colon \Sigma^* \to \mathcal{M}_{(d+1)}(\mathbb{Z})$ such that the last column of $h'(w)$ is $h(w)\mathbf{g}$. Thus (r, w) is the last entry of $\mathbf{s}h'(w)$. Applying the same technique to $(g(w))^{\mathrm{tr}}.\mathbf{s}^{\mathrm{tr}}$, there is a morphism $g\colon \Sigma^* \to \mathcal{M}_{d+2}(\mathbb{Z})$ such that the last component of the last column of $g(w)$ is (r, w), hence $g^{-1}(\mathbb{Z}^{(d+2)^2-1} \times \mathbb{Z}_+) = \mathsf{supp}_+(r)$. By Theorem 4, this shows that $\mathsf{supp}_+(r) \in \mathcal{L}_{\mathrm{DetAPA}}$. $\square$

Remark 2 The class of positive supports of $\mathbb{Z}$-rational series is the class of $\mathbb{Q}$-*stochastic languages* (see, e.g., [28]). We note that the fact that unary $\mathbb{Q}$-stochastic languages are not closed under union [28] implies, as any regular language is $\mathbb{Q}$-stochastic, that there are nonregular unary languages in $\mathcal{L}_{\mathrm{DetAPA}}$. As the unary languages of $\mathcal{L}_{\mathrm{CA}}$ are precisely the regular ones [23], this in turn implies that $\mathcal{L}_{\mathrm{CA}} \subsetneq \mathcal{L}_{\mathrm{DetAPA}}$ over unary languages. The typical shape of these languages is based on having a^n in the language if $\sin(n \times 2\pi\theta) > 0$ with some transcendental number θ.

Expressing $\mathcal{L}_{\mathrm{DetAPA}}$ using linear representations allows for a translation of an expressiveness result of $\mathbb{Q}$-stochastic languages appearing in [28, Theorem III.4.7]. Therein, a certain measure, that we call the *prefix diversity* of a language, is shown to be polynomially bounded for $\mathbb{Q}$-stochastic languages. We conclude this section with the formal definition of this measure and the expressiveness lemma that lifts the aforementioned polynomial property to DetAPA languages.

Definition 2 (Prefix diversity) Let $L \subseteq \Sigma^*$ and write $\chi_L \colon \Sigma^* \to \{0,1\}$ for its characteristic function (i.e., $\chi_L(w) = 1$ iff $w \in L$). The prefix diversity of L is the function $\mathsf{pd}_L \colon \mathbb{Z}_+ \to \mathbb{Z}_+$ defined by:

$$\mathsf{pd}_L(n) = \max_{v_1, v_2, \ldots, v_n \in \Sigma^*} |\{(\chi_L(v_1 v), \chi_L(v_2 v), \ldots, \chi_L(v_n v)) \mid v \in \Sigma^*\}| \enspace .$$

Lemma 6 *Any DetAPA language has a polynomially bounded prefix diversity.*

Proof This statement holds for $\mathbb{Q}$-stochastic languages, see, e.g., [28, Theorem III.4.7]. Let L be a DetAPA language. Now L is the Boolean combination of $\mathbb{Q}$-stochastic languages $L_1, L_2, \ldots, L_k$ and the value of $\chi_L(w)$ depends only on $\chi_{L_1}(w), \chi_{L_2}(w), \ldots, \chi_{L_k}(w)$.

Let $n > 0$ be an integer and $v_1, v_2, \ldots, v_n$ be distinct words. Then:

$$\begin{aligned} &|\{(\chi_L(v_1 v), \chi_L(v_2 v), \ldots, \chi_L(v_n v)) \mid v \in \Sigma^*\}| \\ &\qquad < \prod_{i \in [k]} |\{(\chi_{L_i}(v_1 v), \chi_{L_i}(v_2 v), \ldots, \chi_{L_i}(v_n v)) \mid v \in \Sigma^*\}| \\ &\qquad < \prod_{i \in [k]} \mathsf{pd}_{L_i}(n) \enspace . \end{aligned}$$

This concludes the proof, as the latter product is polynomially bounded. □

6 Algebra, Complexity, and Language Properties

We now provide consequences of the characterizations of Sections 4 and 5, with a special focus on completing our understanding of the class $\mathcal{L}_{\text{DetAPA}}$. To this end, we rely on the property presented in Lemma 6, which we use to show that some languages do not belong to $\mathcal{L}_{\text{DetAPA}}$. The proofs of nonmembership being quite similar in structure, we group them into one proposition. Let b be the infinite word consisting of the concatenation of the binary representations of the positive natural numbers, i.e., $b = 1\,10\,11\,100 \cdots$, and define $L_{\text{bin}} = \{a^i \mid b_i = 1 \wedge i > 0\}$. Independently, let $L_= = \cup_n a^n\# \cdot (a+\#)^* \cdot \#a^n\#$, and define $L_\times = \{a^n b^{nm} c^{\geq m} \mid n, m \in \mathbb{Z}_+\}$.

Proposition 2 *None of* L_{bin}*,* $L_= \cdot (a+\#)^*$*,* $(L_=)^*$*, or* $L_\times$ *are in* $\mathcal{L}_{\text{DetAPA}}$*.*

Proof We rely on Lemma 6 in each case. Suppose to the contrary that L_{bin} (resp. $L_= \cdot (a+\#)^*$, $(L_=)^*$, or $L_\times$) is in $\mathcal{L}_{\text{DetAPA}}$, and write χ for its characteristic function and pd for its polynomially bounded prefix diversity. Pick any n such that $2^n > \mathsf{pd}(n)$. For each language, we successively define $v_1, v_2, \ldots, v_n$, and then for any $r \in \{0,1\}^n$, an additional word v, in such a way that:

$$(\chi(v_1 v), \chi(v_2 v), \ldots, \chi(v_n v)) = r \enspace . \qquad (\star)$$

(For L_{bin}) Let $v_i = a^i$ for any i and $v = a^\ell$ with ℓ the first position in b such that $b_{\ell+1}b_{\ell+2}\cdots b_{\ell+n} = r$;

(For $L_= \cdot (a+\#)^*$ and $(L_=)^*$) Let $v_i = a^i\#$ and $v = (v_{s_1})^2(v_{s_2})^2\cdots(v_{s_k})^2$ where $s_1, s_2, \ldots, s_k$ are the positions at which r is 1.

(For $L_\times$) Let $v_i = a^{p_i}$ and $v = b^{p_{s_1}p_{s_2}\cdots p_{s_k}}c^{p_1p_2\cdots p_k}$ where p_i is the i-th prime number and $s_1, s_2, \ldots, s_k$ are the positions at which r is 1.

In each case, Eq. ($\star$) is easily checked. This in turn implies that:

$$|\{(\chi(v_1v), \chi(v_2v), \ldots, \chi(v_nv)) \mid v \text{ any word}\}| = 2^n > \mathsf{pd}(n) \ .$$

This contradicts the definition of pd, hence L_{bin}, $L_= \cdot (a+\#)^*$, $(L_=)^*$, and $L_\times$ are not in $\mathcal{L}_{\text{DetAPA}}$. $\square$

6.1 On $\mathbf{Z}^+ \circledcirc \mathbf{M}$, $\mathbf{Z}^+ \Box \mathbf{M}$, and $\mathbf{ZMat}^+$

It is easily shown [24] that for any three typed monoids M, N, and N', it holds that:

$$\mathcal{L}((M \Box N) \Box N') \subseteq \mathcal{L}(M \Box (N \Box N'))$$
$$\text{and } \mathcal{L}((M \circledcirc N) \circledcirc N') \subseteq \mathcal{L}(M \circledcirc (N \circledcirc N')) \ .$$

This immediately implies, by Theorem 1, and as $\mathbf{M} \Box \mathbf{M} = \mathbf{M} \circledcirc \mathbf{M} = \mathbf{M}$:

Proposition 3 *The smallest variety containing $\mathbf{Z}^+ \circledcirc \mathbf{M}$ (resp. $\mathbf{Z}^+ \Box \mathbf{M}$, $\mathbf{ZMat}^+ \Box \mathbf{M}$) is closed under wreath (resp. block) product on the right with $\mathbf{M}$ (i.e., if M is in the variety and N is a finite typed monoid, then $M \circledcirc N$ is also in the variety).*

Theorem 6, stating that DetAPA and UnAPA recognize the same languages, has thus the following corollary:

Corollary 2 *The smallest variety containing $\mathbf{ZMat}^+$ is closed under block product with $\mathbf{M}$ on the right.*

We may naturally ask whether these varieties are closed under wreath product with $\mathbf{M}$ on the left. To show this is not the case, let $U_1 = (\{0, 1\}, \times)$, then:

Proposition 4 *The language $L_\times \notin \mathcal{L}_{\text{CA}} \cup \mathcal{L}_{\text{DetAPA}}$ of Proposition 2 is recognized by*[3] *$(U_1)^2 \circledcirc \mathbb{Z}[\mathbb{Z}_+]^d$ for some $d > 0$.*

Proof First, we note that $L_\times \notin \mathcal{L}_{\text{CA}}$ is immediate since $\mathsf{Pkh}(L_\times)$ is not semilinear.

[3] A wreath product with an infinite typed monoid on the right results in an uncountable monoid, an undesirable property that was circumvented in [25]. We note that Theorem 4 stays true with a definition of wreath product mimicking that of [25].

Let us now define a morphism $h\colon \{a,b,c\}^* \to U_1 \circledcirc \mathbb{Z}[\mathbb{Z}_+]^3$ as follows:

$$h(a) = (1, M_a) \text{ with } M_a = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$h(b) = (1, M_b) \text{ with } M_b = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$h(c) = (f_c, M_c) \text{ with } M_c = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\text{and } f_c(M) = 0 \text{ iff } (MM_c)_{1,3} = 0 \ ,$$

where the function 1 is the constant function mapping any matrix to 1. Now for a word $a^i b^j c^k$, it holds that:

$$h(a^i b^j c^k) = (f, M_{i,j,k}) \text{ where } M_{i,j,k} = \begin{pmatrix} 1 & i & j - ik \\ 0 & 1 & -k \\ 0 & 0 & 1 \end{pmatrix} \ .$$

On the other hand, we have:

$$f(1) = f_c(M_{i,j,0}) \times f_c(M_{i,j,1}) \times \cdots \times f_c(M_{i,j,k-1}) \ .$$

This value is 0 precisely when there is a $j' < k$ such that $f_c(M_{i,j,j'}) = 0$. By definition, this is the case if $M_{i,j,j'}M_c = M_{i,j,j'+1}$ has a 0 in the top right corner, that is, if $j = i(j'+1)$. Thus $f(1)$ is 0 iff $j = i \times j'$ for some $j' \geq k$. This shows that $a^*b^*c^* \cap h^{-1}(\{(f, M) \mid f(1) = 0\}) = L_\times$. Augmenting the matrices M_a, M_b, and M_c to check that the input is in $a^*b^*c^*$ is then easy using an extra copy of U_1, showing that $L_\times$ is recognized by $(U_1)^2 \circledcirc \mathbb{Z}[\mathbb{Z}_+]^d$. □

Corollary 3 *None of the smallest varieties containing* $\mathbf{Z}^+ \circledcirc \mathbf{M}$*,* $\mathbf{Z}^+ \,\Box\, \mathbf{M}$*, or* $\mathbf{ZMat}^+$ *are closed under wreath product with* $\mathbf{M}$ *on the left.*

6.2 Complexity of UnCA and DetAPA Languages

We assume some familiarity with the basic notions of circuit complexity and descriptive complexity. We follow the notations of Straubing [29]. For instance, in the formula $\exists x(\exists y(x < y \wedge Q_a x \wedge Q_b y))$, the variables x and y range over the positions in a word, and $Q_a x$ means that there is an a at that position; thus the formula describes the language of words over $\{a, b\}$ that have a b appearing after an a. The main circuit complexity class we will rely on is NC^1, the class of languages recognized by circuits of polynomial size, logarithmic depth, and constant fan-in.

From Theorem 3, we derive a logical characterization of $\mathcal{L}_{\mathrm{UnCA}}$ and deduce a complexity upper bound for it. Let MSO[<] be the monadic second-order logic with $<$ as the unique numerical predicate, a logic that expresses exactly the regular languages [8]. Now define the *extended majority* quantifier $\widehat{\mathrm{Maj}}$, introduced in [5], as:

$$w \models \widehat{\mathrm{Maj}}\ \ x\ \ \langle\phi_i\rangle_{i=1,\ldots,m} \text{ iff } \sum_{j\in[|w|]} |\{i \mid w_{x=j} \models \phi_i\}| - |\{i \mid w_{x=j} \not\models \phi_i\}| > 0\ \ .$$

Further, let $\mathsf{B}(\widehat{\mathrm{Maj}} \circ \mathrm{MSO}[<])$ be the set of formulas that are Boolean combinations of formulas of the form:

$$\widehat{\mathrm{Maj}}\ \ x\ \ \langle\phi_i\rangle_{i=1,\ldots,m}\ \ ,$$

where each ϕ_i is an MSO[$<$] formula. Then:

Theorem 8 *A language is that of an UnCA if and only if it is expressible as a* $\mathsf{B}(\widehat{\mathrm{Maj}} \circ \mathrm{MSO}[<])$*-formula. Hence,* $\mathcal{L}_{\mathrm{UnCA}} \subsetneq \mathrm{NC}^1$.

Proof We first show that the languages recognized by $\mathbb{Z}[\mathbb{Z}_+]$ are those expressible as a formula of the form (or negation of) $\widehat{\mathrm{Maj}}\ \ x\langle Q_{A_i}x\rangle_{i=1,\ldots,m}$ where $A_i \subseteq \Sigma$, and $Q_{A_i}x$ is short for $\bigvee_{a\in A_i} Q_a x$.

Let $L \in \mathcal{L}(\mathbb{Z}[\mathbb{Z}_+])$, i.e., let $h\colon \Sigma^* \to \mathbb{Z}$ be a morphism and suppose $L = h^{-1}(\mathbb{Z}_+)$ (if $L = h^{-1}(\overline{\mathbb{Z}_+})$, then the negation of the formula we obtain here will describe L). We suppose moreover, w.l.o.g., that each $h(a)$, $a \in \Sigma$, is even. Now let m be $\max\{|h(a)| \mid a \in \Sigma\}$ and define, for $1 \le i \le m$:

$$A_i = \{a \in \Sigma \mid m + h(a) \ge 2 \times i\}\ \ .$$

Now let $w \in \Sigma^*$ be a word and $1 \le x \le |w|$. Then it holds that:

$$h(w_x) = \underbrace{|\{i \mid w_x \in A_i\}|}_{(m+h(a))/2} - \underbrace{|\{i \mid w_x \notin A_i\}|}_{m-(m+h(a))/2}\ \ .$$

Thus for $w \in \Sigma^*$, $h(w) > 0$ iff $w \models \widehat{\mathrm{Maj}}\ \ x\langle Q_{A_i}x\rangle_{i=1,\ldots,m}$, thus the language expressed by this latter formula is $h^{-1}(\mathbb{Z}_+) = L$.

Conversely, consider a formula $\widehat{\mathrm{Maj}}\ \ x\langle Q_{A_i}x\rangle_{i=1,\ldots,m}$. Then let $h\colon \Sigma^* \to \mathbb{Z}$ be the morphism defined by $h(a) = |\{i \mid a \in A_i\}| - |\{i \mid a \notin A_i\}|$, for $a \in \Sigma$. We have that for $w \in \Sigma^*$, $h(w) > 0$ iff the formula under consideration holds true, implying that the language recognized by the formula is $h^{-1}(\mathbb{Z}_+)$.

It follows that the languages recognized by $\mathbf{Z}^+$ are the Boolean combinations of languages expressible as such formulas. Now the languages (with one free variable) recognized by finite monoids are those recognized by MSO[$<$] formulas. Thus the *block product principle* [24, Theorem 3.40] implies that the languages of $\mathcal{L}_{\mathrm{UnCA}} = \mathcal{L}(\mathbf{Z}^+ \,\square\, \mathbf{M})$ are those expressible as Boolean combinations of formulas of the form of the statement of the lemma. Similarly, the regular languages (with one free variable) are recognized by NC^1 circuits, and a formula or negation of a formula of the form $\widehat{\mathrm{Maj}}\ \ x\langle Q_{A_i}x\rangle_{i=1,\ldots,m}$ can be expressed by a threshold circuit. Now [24, Lemma 4.29] implies that $\mathcal{L}_{\mathrm{UnCA}} \subseteq \mathrm{NC}^1$. Strictness is implied by Theorem 4. $\square$

Remark 3 The (non)closure properties observed in Section 6.1 can be interpreted, thanks to [24, Chapter 4], in terms of expressiveness of some logics. Proposition 3 is equivalent to the following (trivial) statement: Replacing a subformula of a $\mathsf{B}(\widehat{\mathrm{Maj}} \circ \mathrm{MSO}[<])$-formula by an MSO[<] formula preserves the fact that the language is expressed by a $\mathsf{B}(\widehat{\mathrm{Maj}} \circ \mathrm{MSO}[<])$-formula. Proposition 4 implies that there is a $\mathsf{B}(\widehat{\mathrm{Maj}} \circ \mathrm{MSO}[<])$-formula $\phi(x, y)$ such that $\exists x \exists y(\phi(x, y))$ is not expressible as a $\mathsf{B}(\widehat{\mathrm{Maj}} \circ \mathrm{MSO}[<])$-formula.

Let $\#\mathrm{NC}^1$ be the class of functions computed by DLOGTIME-uniform arithmetic circuits over $\{+, \times\}$ of polynomial size and logarithmic depth and PNC^1 be the class of languages expressible as $\{w \mid f(w) > 0\}$ for $f \in \#\mathrm{NC}^1$ (see [12]). Note that PNC^1 is included in deterministic logspace. As iterated matrix multiplication can be done in $\#\mathrm{NC}^1$ and PNC^1 is closed under the Boolean operations, it is readily seen from Theorem 7 (and Proposition 2 for the strictness) that:

Corollary 4 $\mathcal{L}_{\mathrm{DetAPA}} \subsetneq \mathrm{PNC}^1$.

6.3 New Expressiveness and Nonclosure Properties of DetAPA

This section relies on Proposition 2 to complete our understanding of $\mathcal{L}_{\mathrm{DetAPA}}$. Although $\mathcal{L}_{\mathrm{DetAPA}} \subseteq \mathcal{L}_{\mathrm{APA}}$ is immediate, it was not known whether the two classes differ; we show that they do, in particular over unary languages:

Theorem 9 *There are unary languages in $\mathcal{L}_{\mathrm{APA}}$ which are not in $\mathcal{L}_{\mathrm{DetAPA}}$.*

Proof We show that the language L_{bin} from Proposition 2, that we know lies outside $\mathcal{L}_{\mathrm{DetAPA}}$, is an APA language.

We first note that a DetAPA can store the integer value of its binary input into a register. Indeed, reading $b \in \{0, 1\}$, a register x can be updated with $2 \times x + b$, and at the end of the computation, x will hold the correct value.

Let $\Sigma = \{0, 1\} \times (\{0, 1\} \cup \{0, 1\}^2)$. The *inc-representation* of a number $n > 0$ is the following word over Σ^*:

$$\begin{pmatrix} b_1 \\ b'_0 b'_1 \end{pmatrix} \begin{pmatrix} b_2 \\ b'_2 \end{pmatrix} \cdots \begin{pmatrix} b_k \\ b'_k \end{pmatrix} \in (\{0, 1\} \times \{0, 1\}^2)(\{0, 1\} \times \{0, 1\})^* \ ,$$

with $b_1 b_2 \cdots b_k$ the binary representation of n with a leading 1 and $b'_0 b'_1 b'_2 \cdots b'_k$ a binary representation of $n + 1$. Define then L' as the set of words $w_1 w_2 \cdots w_n$ where each w_i is the inc-representation of i, for all $n > 0$.

Assume $L' \in \mathcal{L}_{\mathrm{DetAPA}}$, we first show how this allows us to show that $L_{\mathrm{bin}} \in \mathcal{L}_{\mathrm{APA}}$. With the w_i's as above, let L'' be the language of words of the form $w_1 w_2 \cdots w_n u$ where u is a prefix of the binary expansion of $n + 1$; we show $L'' \in \mathcal{L}_{\mathrm{APA}}$. The APA for L'' runs the DetAPA for L', and guesses nondeterministically that it is reading w_n, the last of the inc-representations. It then stores the integer value of the binary encoding $b'_0 b'_1 \cdots b'_\ell$ into an additional register x, where ℓ is a guessed value. Thus x contains the integer value of a prefix of the binary value of $n + 1$. After reading w_n, the APA continues by reading over the alphabet $\{0, 1\}$ the binary expansion of a value y, with a leading 1, then accepts if $x = y$. Thus this suffix is accepted iff it is a prefix of the binary expansion of $n + 1$, and the language recognized is L''. The language L_{bin} is then obtained as the morphic image of $L'' \cap \Sigma^*\{0, 1\}^*1$ under the length-preserving morphism mapping every letter to a. This shows that $L_{\text{bin}} \in \mathcal{L}_{\text{APA}}$ by the closure properties of APA [9].

To show that $L' \in \mathcal{L}_{\text{DetAPA}}$, we rely on a technique that is already exploited in [9, Lemma 4.13]: a DetAPA can check an unbounded number of times that some registers are zero, in such a way that an extra register holds 0 iff all the tests succeeded. To do so, the DetAPA is equipped with the extra register x and maintains the property that x is either zero or greater in absolute value than any other registers. The register x is modified on two occasions: 1. When a register y is being tested for equality with 0, x is updated with $x + y$; 2. At each step of the execution of the automaton, x is multiplied by a large constant to preserve the aforementioned property. 1 and 2 combined ensure that x is zero iff all the tested values y were 0. In a similar way, a DetAPA can check an unbounded number of times that two registers are equal.

The DetAPA for L' then works as follows. It initializes a register c to 1. Then it reads a word $w \in \binom{1}{\{0,1\}\{0,1\}}\binom{\{0,1\}}{\{0,1\}}^*$ that cannot be extended to the right with a letter from $\binom{\{0,1\}}{\{0,1\}}$, and checks that the integer value of the first component of w is c, and the second component $c + 1$. It then loops back to the reading the next w, while incrementing c, and accepts if all the tests succeeded. □

Over unary languages, $\mathcal{L}_{\text{CA}} = \mathcal{L}_{\text{DetCA}} \subsetneq \mathcal{L}_{\text{DetAPA}}$ (the equality coming from [23], and the strict inclusion from Remark 2). Over larger alphabets, it is known that $\mathcal{L}_{\text{CA}} \not\subseteq \mathcal{L}_{\text{DetAPA}}$ [9, Proposition 28], but the reverse noninclusion was left open. We show:

Proposition 5 *There are languages in $\mathcal{L}_{\text{CA}}$ which are not in $\mathcal{L}_{\text{DetAPA}}$.*

Proof The language $L_= \cdot \#(a + \#)^*$ of Proposition 2 is not in $\mathcal{L}_{\text{DetAPA}}$, but it is in $\mathcal{L}_{\text{CA}}$: the automaton simply counts the number n of a at the beginning of the word, and guesses a position at which the second a^n appears. □

It is now possible to state the precise relationship among all the classes studied here and in previous works, and this is presented in Fig. 1.

For completeness, we remark that over *bounded languages*,[4] a well-behaved class that has received continuous attention since the seminal work of Ginsburg and Spanier [17], the following chain holds:

$$\text{REG} \subsetneq \mathcal{L}_{\text{DetCA}} = \mathcal{L}_{\text{UnCA}} = \mathcal{L}_{\text{CA}} \subsetneq \mathcal{L}_{\text{DetAPA}} = \mathcal{L}_{\text{UnAPA}} \subsetneq \mathcal{L}_{\text{APA}} \quad ,$$

the CA results originating from [10].

[4] A language L is *bounded* if $L \subseteq w_1^* w_2^* \cdots w_k^*$ for some words $w_1, w_2, \ldots, w_k$.

The landscape of closure properties of $\mathcal{L}_{\text{DetAPA}}$ was also left with some holes in previous works. We thus complete [9, Figure 1], refined in [11], by showing:

Theorem 10 $\mathcal{L}_{\text{DetAPA}}$ *is not closed under concatenation with regular languages, starring, and commutative closure.*

Proof This is a consequence of Proposition 2. Indeed, the language $L_=$ is in $\mathcal{L}_{\text{UnCA}}$, as an automaton can unambiguously guess that it is reading the last block of a's, hence in $\mathcal{L}_{\text{DetAPA}}$. However, $L_= \cdot \#(a + \#)^* \notin \mathcal{L}_{\text{DetAPA}}$, hence $\mathcal{L}_{\text{DetAPA}}$ is not closed under concatenation. Similarly, $(L_=)^* \notin \mathcal{L}_{\text{DetAPA}}$, hence $\mathcal{L}_{\text{DetAPA}}$ is not closed under starring. Finally, it is not hard to see that the language $L = \{a^n c^m b^{mn} c^* \mid m, n \in \mathbb{Z}_+\}$ is a DetAPA language. However, $\mathsf{Comm}(L) \cap a^*b^*c^*$ is $L_\times \notin \mathcal{L}_{\text{DetAPA}}$, thus $\mathsf{Comm}(L) \notin \mathcal{L}_{\text{DetAPA}}$. $\square$

Remark 4 Similarly, the DetAPA language $L = \{a^m\#b^n\#c^{mn} \mid m, n \in \mathbb{N}\}$ could be shown to verify $a^* \cdot L \notin \mathcal{L}_{\text{DetAPA}}$, thus $\mathcal{L}_{\text{DetAPA}}$ is not even closed under concatenation with regular *unary* languages.

7 Conclusion

Connections between variants of the Parikh automaton and different algebraic formalisms were investigated. As a main consequence of this study, we completed our knowledge of the interrelationships and closure properties of the language classes that arise. These properties are summed up in Figs. 1 and 2.

Further, natural characterizations of the language classes defined by deterministic and unambiguous constrained automata, in the theory of typed monoids, were obtained. Given the tight link between typed monoids and circuit complexity, we hope that these characterizations will suggest refinements that help to better understand the classes PNC^1 and NC^1.

An additional characterization of one of our central classes of focus, $\mathcal{L}_{\text{DetAPA}}$, relies on formula power series and may further shed light on $\mathbb{Q}$-stochastic languages. In particular, DetAPA may offer a different perspective on recent developments in the study of unary $\mathbb{Q}$-stochastic languages [2]. Independently, the notion of prefix diversity (Definition 2), singled out as a central tool to show nonmembership, could play an important role in the study of the complexity of these unary languages. As such, studying this measure may be a worthwhile endeavor. A striking shortcoming of this measure, as noted by Turakainen [30], is that it is *linear* of the Dyck language over $\{a, \dot{a}\}$, but *exponential* of $D_1 \cdot a(a + \dot{a})^*$, although we conjecture that $D_1 \notin \mathcal{L}_{\text{DetAPA}}$.

Bridging questions on circuits and on unary languages, we note that the unary languages in $\mathcal{L}_{\text{DetAPA}}$, and indeed the bounded languages in $\mathcal{L}_{\text{DetAPA}}$, can be shown to belong to the DLOGTIME-DCL-uniform variant of NC^1. Recall that the latter is not

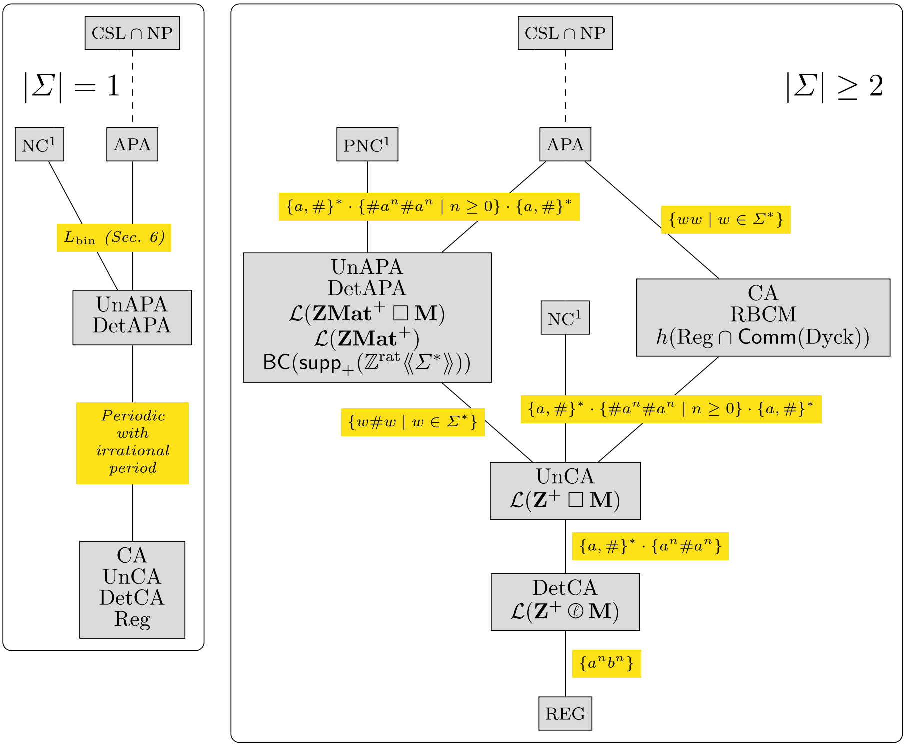

Fig. 1 Class relationships. Left: over a unary alphabet. Right: over a nonunary alphabet. Classes in the same box are equal. Dashed lines indicate inclusions, bottom to top. Solid lines denote strict inclusions, and witnesses are given on the line

known to equal what is commonly referred to as DLOGTIME-uniform NC^1 (see [31, p. 162]), or ALOGTIME. Yet we were unable to show that the unary languages in $\mathcal{L}_{\text{DetAPA}}$ belong to the latter—do they? An intriguing related question is whether the language L_{fib} defined as $\{a^n \mid n \text{ is a Fibonacci number}\}$ belongs to $\mathcal{L}_{\text{DetAPA}}$; it is indeed possible to show [32] that L_{fib} is the positive support of a rational $\mathbb{R}$-series—as opposed to a $\mathbb{Z}$-series—but we conjecture that $L_{\text{fib}} \notin \mathcal{L}_{\text{DetAPA}}$.

	$\cup$	$\cap$	$-$	$\cdot$	h	$h_{\not\varepsilon}$	h^{-1}	Comm	L^*	L^{-1}	L^{R}
DetCA	Y	Y	Y	N	N	N	Y	Y	N	Y	N
UnCA	Y	Y	Y	N	N	N	Y	Y	N	Y	Y
CA	Y	Y	N	Y	Y	Y	Y	Y	N	Y	Y
DetAPA	Y	Y	Y	***N***	N	***N***	Y	***N***	***N***	N	Y
APA	Y	Y	N^1	Y	N	Y	Y	Y	Y	N	Y

Fig. 2 Closure properties (union, intersection, complement, concatenation, morphisms, nonerasing morphisms, inverse morphism, commutative closure, starring, quotient, reversal). In bold, properties proven here, the others being found in [9, 23]. ^{1}Assuming EXP $\neq$ NEXP

Acknowledgements We thank Michael Blondin and Charles Paperman for comments on early versions of this article. Part of this work was done during the Dagstuhl Seminar 15401 "Circuits, Logic and Games."

References

1. Abdulla, P.A., Atig, M.F., Meyer, R., Salehi, M.S.: What's Decidable about Availability Languages. In: FSTTCS, LIPIcs, vol. 45, pp. 192–205. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2015). https://doi.org/10.4230/LIPIcs.FSTTCS.2015.192
2. Akshay, S., Genest, B., Karelovic, B., Vyas, N.: On Regularity of Unary Probabilistic Automata. In: STACS, LIPIcs, vol. 47, pp. 8:1–8:14. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik (2016). https://doi.org/10.4230/LIPIcs.STACS.2016.8
3. Barrington, D.A.M.: Bounded-width polynomial size branching programs recognize exactly those languages in NC^1. J. Comput. Syst. Sci. **38**, 150–164 (1989)
4. Barrington, D.A.M., Thérien, D.: Finite monoids and the fine structure of NC^1. J. Assoc. Comput. Mach. **35**, 941–952 (1988)
5. Behle, C., Krebs, A., Mercer, M.: Linear Circuits, Two-Variable Logic and Weakly Blocked Monoids. In: Mathematical Foundations of Computer Science, LNCS, vol. 4708, pp. 147–158. Springer, Berlin (2007)
6. Behle, C., Krebs, A., Reifferscheid, S.: Typed Monoids - an Eilenberg-Like Theorem for Non Regular Languages. In: Algebraic Informatics, LNCS, vol. 6742, pp. 97–114. Springer, Berlin (2011)
7. Berstel, J., Reutenauer, C.: Noncommutative Rational Series with Applications. Encyclopedia of Mathematics and its Applications. Cambridge University Press, Cambridge (2010)
8. Bu̇chi, J.R.: Weak second order arithmetic and finite automata. Z. Math. Logik Grundlagen Math. **6**, 66–92 (1960)
9. Cadilhac, M., Finkel, A., McKenzie, P.: Affine Parikh automata. RAIRO - Theoretical Informatics and Applications **46**(04), 511–545 (2012). https://doi.org/10.1051/ita/2012013
10. Cadilhac, M., Finkel, A., McKenzie, P.: Bounded Parikh automata. Int. J. Found. Comput. Sci. **23**(08), 1691–1709 (2012)
11. Cadilhac, M., Finkel, A., McKenzie, P.: Unambiguous constrained automata. Int. J. Found. Comput. Sci. **24**(7), 1099–1116 (2013). https://doi.org/10.1142/S0129054113400339
12. Caussinus, H., McKenzie, P., Thérien, D., Vollmer, H.: Nondeterministic NC^1 computation. J. Comput. Syst. Sci. **57**, 200–212 (1998)
13. Chomsky, N., Schützenberger, M.P.: The Algebraic Theory of Context-Free Languages. In: Computer Programming and Formal Systems, Studies in Logic and the Foundations of Mathematics, vol. 26, pp. 118–161. Elsevier, Amsterdam (1959). https://doi.org/10.1016/S0049-237X(09)70104-1
14. Eilenberg, S.: Automata, Languages, and Machines, Volume B. Pure and Applied Mathematics. Academic Press, Cambridge (1976)
15. Enderton, H.B.: A mathematical introduction to logic. Academic Press, Cambridge (1972)
16. Figueira, D., Libkin, L.: Path Logics for Querying Graphs: combining Expressiveness and Efficiency. In: LICS, pp. 329–340 (2015). https://doi.org/10.1109/LICS.2015.39
17. Ginsburg, S., Spanier, E.H.: Bounded regular sets. Proc. Am. Math. Soc. **17**(5), 1043–1049 (1966). https://doi.org/10.2307/2036087
18. Ginsburg, S., Spanier, E.H.: Semigroups, Presburger formulas and languages. Pac. J. Math. **16**(2), 285–296 (1966)
19. Hoenicke, J., Meyer, R., Olderog, E.R.: Kleene, Rabin, and Scott are Available. In: Proceedings of the 21St International Conference on Concurrency Theory, CONCUR'10, pp. 462–477. Springer, Berlin (2010)
20. Ibarra, O.H.: Reversal-bounded multicounter machines and their decision problems. J. ACM **25**(1), 116–133 (1978). https://doi.org/10.1145/322047.322058
21. Kambites, M.: Formal languages and groups as memory. Commun. Algebra **37**(1), 193–208 (2009). https://doi.org/10.1080/00927870802243580
22. Karianto, W.: Parikh automata with pushdown stack. Diploma thesis RWTH Aachen (2004)
23. Klaedtke, F., Rueß, H.: Monadic second-order logics with cardinalities. In: International Colloquium on Automata, Languages and Programming, LNCS, vol. 2719, pp. 681–696. Springer, Berlin (2003). https://doi.org/10.1007/3-540-45061-0_54

24. Krebs, A.: Typed Semigroups, Majority Logic, and Threshold Circuits. Ph.D. thesis, Eberhard Karls University of Tübingen (2008)
25. Krebs, A., Lange, K.J., Reifferscheid, S.: Characterizing TC^0 in terms of infinite groups. Theory of Computing Systems **40**(4), 303–325 (2007)
26. Parikh, R.J.: On context-free languages. J. ACM **13**(4), 570–581 (1966). https://doi.org/10.1145/321356.321364
27. Pin, J.E.: Mathematical foundations of automata theory (2016). https://www.irif.fr/~jep/PDF/MPRI/MPRI.pdf
28. Salomaa, A., Soittola, M.: Automata-Theoretic Aspects of Formal Power Series. Springer, New York (1978)
29. Straubing, H.: Finite Automata, Formal Logic, and Circuit Complexity. Birkäuser, Boston 1994. https://doi.org/10.1007/978-1-4612-0289-9
30. Turakainen, P.: Some closure properties of the family of stochastic languages. Inf. Control. **18**(3), 253–256 (1971)
31. Vollmer, H.: Introduction to circuit complexity. Springer, Berlin (1999)
32. Zare, D.: For a linear recurrence sequence $(u_n)_{n\geq 0}$, can $\{i|u_i > 0\}$ be the set of Fibonacci numbers? MathOverflow (2015). http://mathoverflow.net/q/222379