Electronic Colloquium on Computational Complexity, Report No. 145 (2015)

Arithmetic circuit classes over $\mathbb{Z}_m$

Eric Allender
Department of Computer Science, Rutgers University
Piscataway, NJ, USA
allender@cs.rutgers.edu

Asa Goodwillie
Department of Mathematics, Amherst College
Amherst, MA, USA
asa.k.goodwillie@gmail.com

September 4, 2015

Abstract

We continue the study of the complexity classes $\mathsf{VP}(\mathbb{Z}_m)$ and $\Lambda\mathsf{P}(\mathbb{Z}_m)$ which was initiated in [AGM15]. We distinguish between "strict" and "lax" versions of these classes and prove some new equalities and inclusions between these arithmetic circuit classes and various subclasses of ACC^1.

1 Introduction

This short note considers the complexity classes VP and $\Lambda\mathsf{P}$, continuing an investigation that was begun in [AGM15]. VP is a relatively well-studied complexity class (see, e.g. [Val79, VSBR83, Bür99, Bür00, GW96, KP11]), while the "dual" notion $\Lambda\mathsf{P}$ is much less familiar. We briefly review these classes, before stating the contributions of this paper.

VP is usually studied in the context of algebraic complexity theory, as a class of polynomials. Given any semiring R, $\mathsf{VP}(R)$ is the class of multivariate polynomials of polynomially-bounded degree that can be represented by a family of arithmetic circuits $\{C_n : n \in \mathbb{N}\}$ over R, where each circuit C_n has size polynomial in n. It is known [AJMV98] that an equivalent characterization of $\mathsf{VP}(R)$ can be obtained by additionally imposing the restriction that each C_n have depth $O(\log n)$, where each $+$ gate has unbounded fan-in, and each $\times$ gate has fan-in two. (Such circuits are said to have *semiunbounded* fan-in.)

An investigation was begun in [AGM15] into the *dual* notion of semiunbounded fan-in arithmetic circuits, where the $\times$ gates have unbounded fan-in, the $+$ gates have fan-in two, while the size and depth are still restricted to be

ISSN 1433-8092

polynomial and logarithmic in n, respectively. This class of polynomials was dubbed $\mathsf{\Lambda P}(R)$ [AGM15].

Here, as in [AGM15], our focus is on how these classes relate to various well-known subclasses of P. One natural way to do this is to consider the class of languages that are logspace-Turing-reducible to $\mathsf{VP}(R)$ and $\mathsf{\Lambda P}(R)$, in the following sense:

Let f be a function, $f : \{0,1\}^* \to \{0,1\}^*$ where the length of $f(x)$ is bounded by a polynomial in the length of x. L^f is the class of languages accepted by logspace-bounded oracle Turing machines equipped with a query tape that is not subject to the space bound, using the oracle $\{(x, i, b) :$ the i-th bit of $f(x)$ is $b\}$. For more formal definitions, see, e.g. [LL76]. For a class of functions $\mathcal{C}$ (such as $\mathsf{VP}(\mathbb{Q}), \mathsf{VP}(\mathbb{F}_p), \mathsf{\Lambda P}(\mathbb{F}_p), \mathsf{VP}(\mathbb{Z}_m)$, etc.) $\mathsf{L}^{\mathcal{C}}$ is defined to be the union over all $f \in \mathcal{C}$ of L^f (using a natural encoding of elements of $\mathbb{Q}, \mathbb{F}_p$, and $\mathbb{Z}_m$ as binary strings). This gives a useful characterization of the computational complexity of functions in these various classes $\mathcal{C}$.

It is observed in [AGM15] that for any prime p and any positive integer k, $\mathsf{L}^{\mathsf{VP}(\mathbb{F}_{p^k})}$ and $\mathsf{L}^{\mathsf{\Lambda P}(\mathbb{F}_{p^k})}$ each correspond to the class of languages whose characteristic functions lie in $\mathsf{VP}(\mathbb{F}_{p^k})$ and $\mathsf{\Lambda P}(\mathbb{F}_{p^k})$, respectively. That is, for example, if $A \in \mathsf{L}^{\mathsf{\Lambda P}(\mathbb{F}_{p^k})}$, then there is a family of multivariate polynomials $\{f_n : n \in \mathbb{N}\} \in \mathsf{\Lambda P}(\mathbb{F}_{p^k})$, such that for any string $x = x_1 \dots x_n$ of length n, $x \in A$ if and only if $f_n(x_1, \dots, x_n) = 1$, and otherwise $f_n(x_1, \dots, x_n) = 0$. Thus the following convention was adopted in [AGM15]: When it results in no confusion, $\mathsf{VP}(\mathbb{F}_{p^k})$ and $\mathsf{\Lambda P}(\mathbb{F}_{p^k})$ will refer to the class of *languages* whose characteristic functions lie in the given class.

The phrase "When it results in no confusion" can be dangerous. For example, consider the following conjecture that appears in [AGM15]:

Conjecture 1. $\mathsf{ACC}^1 = \bigcup_m \mathsf{L}^{\mathsf{VP}(\mathbb{Z}_m)}$.

The authors of [AGM15] intended that, in this context, $\mathsf{VP}(\mathbb{Z}_m)$ would represent the class of *functions* represented by VP polynomials operating over $\mathbb{Z}_m$. But the notation does not make this clear; one could also attempt to interpret $\mathsf{L}^{\mathsf{VP}(\mathbb{Z}_m)}$ as denoting the class of languages logspace-Turing-reducible to a language whose characteristic function lies in $\mathsf{VP}(\mathbb{Z}_m)$. In this paper, we show that interpreting the notation in this way causes the statement of Conjecture 1 to be rather unlikely to be true.

In order to disambiguate the notation, let us distinguish between the following two classes:

- $\mathsf{strictVP}(\mathbb{Z}_m)$ denotes the class of languages A for which there exists a family of multivariate polynomials $\{f_n : n \in \mathbb{N}\} \in \mathsf{VP}(\mathbb{Z}_m)$, such that for any string $x = x_1 \dots x_n$ of length n, $x \in A$ if and only if $f_n(x_1, \dots, x_n) = 1$, and otherwise $f_n(x_1, \dots, x_n) = 0$. (This corresponds to what is called *exact representation* in the survey by Williams [Wil14].)
- $\mathsf{laxVP}(\mathbb{Z}_m)$ denotes the class of languages A for which there exists a family of multivariate polynomials $\{f_n : n \in \mathbb{N}\} \in \mathsf{VP}(\mathbb{Z}_m)$, such that for any

string $x = x_1 \dots x_n$ of length n, $x \in A$ if and only if $f_n(x_1, \dots, x_n) \neq 0$. (This is essentially the type of representation that is considered in [BBR94].)

The classes $\mathsf{strict}\Lambda\mathsf{P}(\mathbb{Z}_m)$ and $\mathsf{lax}\Lambda\mathsf{P}(\mathbb{Z}_m)$ are defined analogously. For the rest of the paper, the notation $\mathsf{VP}(\mathbb{Z}_m)$ and $\Lambda\mathsf{P}(\mathbb{Z}_m)$ will be used to denote the *function* classes. When working over finite *fields*, we will continue to use the notation $\mathsf{VP}(\mathbb{F}_{p^k})$ and $\Lambda\mathsf{P}(\mathbb{F}_{p^k})$ to denote the language classes, since these language classes capture all of the relevant aspects of the complexity of the function classes, and hence there is little to be gained by using the more cumbersome notation.

1.1 Our Contributions

First, we observe that the language classes laxVP and $\mathsf{lax}\Lambda\mathsf{P}$ capture the computational complexity of the function classes VP and $\Lambda\mathsf{P}$. Namely, we show that the following equalities hold:

- $\mathsf{L}^{\mathsf{VP}(\mathbb{Z}_m)} = \mathsf{L}^{\mathsf{laxVP}(\mathbb{Z}_m)}$.
- $\mathsf{L}^{\Lambda\mathsf{P}(\mathbb{Z}_m)} = \mathsf{L}^{\mathsf{lax}\Lambda\mathsf{P}(\mathbb{Z}_m)}$.

If m is prime, then these classes coincide with $\mathsf{strictVP}(\mathbb{F}_m)$ and $\mathsf{strict}\Lambda\mathsf{P}(\mathbb{F}_m)$, respectively.

The following proposition is also clear:

Proposition 1. *Let p be a prime divisor of m. Then $\mathsf{VP}(\mathbb{F}_p) \subseteq \mathsf{laxVP}(\mathbb{Z}_m)$ and $\Lambda\mathsf{P}(\mathbb{F}_p) \subseteq \mathsf{lax}\Lambda\mathsf{P}(\mathbb{Z}_m)$.*

To see this, let $\{C_n \mid n \in \mathbb{N}\}$ be a family of circuits defining a language in $\mathsf{VP}(\mathbb{F}_p)$. (The argument is identical for $\Lambda\mathsf{P}(\mathbb{F}_p)$.) Then the circuit $(m/p) \times C_n$ evaluates to 0 (mod m) on input x if and only if x is rejected by C_n.

Thus $\mathsf{laxVP}(\mathbb{Z}_m)$ contains the *union* of all $\mathsf{VP}(\mathbb{F}_p)$ for all primes p that divide m. (A similar result holds for the $\Lambda\mathsf{P}$ classes.)

The main contribution of this note is to observe that $\mathsf{strictVP}(\mathbb{Z}_m)$ and $\mathsf{strict}\Lambda\mathsf{P}(\mathbb{Z}_m)$ correspond exactly to the *intersection* of the corresponding classes for each p that divides m.

In Section 5, we give a somewhat better upper bound on the complexity of $\mathsf{L}^{\Lambda\mathsf{P}(\mathbb{Z}_m)}$ than was presented in [AGM15], where it was merely shown that this class is contained in ACC^1.

2 Background

In [AGM15], Allender et al. defined $\Lambda\mathsf{P}$ as a dual counterpart to the older and better known VP class, and established Boolean characterizations of both arithmetic classes over finite fields, showing that for every prime p and every positive integer k, $\mathsf{VP}(\mathbb{F}_{p^k}) = \mathsf{VP}(\mathbb{F}_p) = \mathsf{CC}^1[p]$ and $\Lambda\mathsf{P}(\mathbb{F}_{p^k}) = \mathsf{AC}^1[\mathsf{Supp}(p^k - 1)]$, where $\mathsf{Supp}(m)$ denotes the set of prime factors of a positive integer m.

Since $\mathbb{Z}_p = \mathbb{F}_p$ for any prime p, these results give Boolean characterizations of VP and $\Lambda\mathsf{P}$ over $\mathbb{Z}_p$ when p is prime: $\mathsf{strictVP}(\mathbb{Z}_p) = \mathsf{laxVP}(\mathbb{Z}_p) = \mathsf{CC}^1[p]$ and $\mathsf{strict\Lambda P}(\mathbb{Z}_p) = \mathsf{lax\Lambda P}(\mathbb{Z}_p) = \mathsf{AC}^1[\mathsf{Supp}(p-1)]$. Focusing on the strict versions of VP and $\Lambda\mathsf{P}$, we extend this from prime moduli to moduli that are powers of primes, obtaining Boolean characterizations of $\mathsf{strictVP}$ and $\mathsf{strict\Lambda P}$ over $\mathbb{Z}_{p^k}$, and we describe $\mathsf{strictVP}(\mathbb{Z}_m)$ and $\mathsf{strict\Lambda P}(\mathbb{Z}_m)$ for an arbitrary modulus m in terms of the special prime power cases.

First, we restate the central definitions.

Definition 1. *Let R be a commutative semiring with unity. (Throughout, every (semi)ring we discuss will be assumed to be a (semi)ring with unity, and a homomorphism $\psi : R \to S$ will be assumed to take 1_R to 1_S.) We let $\mathsf{strictVP}(R)$ (resp. $\mathsf{laxVP}(R)$) denote the class of all languages $A \subseteq \{0,1\}^*$ for which there exists a logspace-uniform circuit family $\{C_n \mid n \in \mathbb{N}\}$ such that*

- *the depth of C_n is $O(\log n)$,*
- *each C_n consists of n input gates, along with $\times$ gates of fan-in two and $+$ gates of unbounded fan-in and perhaps some constant gates, each outputting some constant $c \in R$, and*
- *for every string $w = w_1 \cdots w_n \in \{0,1\}^*$ of length n, the output of C_n evaluated over R on input $w_1, \ldots, w_n$ is 1 (resp. non-zero) if $w \in A$ and 0 otherwise.*

To evaluate C_n over R on input $w = w_1 \cdots w_n \in \{0,1\}^*$, we define $v_i = 0_R$ if $w_i = 0$ and $v_i = 1_R$ if $w_i = 1$ (where 0_R and 1_R denote the additive and multiplicative identities of R, respectively) and evaluate C_n on inputs $v_1, \ldots, v_n$.

The definition of $\mathsf{strict\Lambda P}(R)$ and $\mathsf{lax\Lambda P}(R)$ is precisely the same, except that the $+$ gates are restricted to have fan-in two while the fan-in of the $\times$ gates is unrestricted.

This paper should be viewed as a companion to [AGM15]. The reader is referred to [AGM15] for definitions of classes (such as $\mathsf{CC}^1[m]$) that are not defined here.

3 Lax Classes

In this section, we observe that $\mathsf{laxVP}(\mathbb{Z}_m)$ and $\mathsf{lax\Lambda P}(\mathbb{Z}_m)$ capture the complexity of the function classes $\mathsf{VP}(\mathbb{Z}_m)$ and $\Lambda\mathsf{P}(\mathbb{Z}_m)$, respectively.

Proposition 2. *The following equalities hold:*

- $\mathsf{L}^{\mathsf{VP}(\mathbb{Z}_m)} = \mathsf{L}^{\mathsf{laxVP}(\mathbb{Z}_m)}$.
- $\mathsf{L}^{\Lambda\mathsf{P}(\mathbb{Z}_m)} = \mathsf{L}^{\mathsf{lax\Lambda P}(\mathbb{Z}_m)}$.

If m is prime, then these classes coincide with $\mathsf{strictVP}(\mathbb{F}_m)$ and $\mathsf{strict\Lambda P}(\mathbb{F}_m)$, respectively.

Proof. In each case, containment from right to left is immediate. Thus let A be an element of L^f for some $f \in \mathsf{VP}(\mathbb{Z}_m)$. Let g be the function $f - x$ for a new variable x. Clearly $g \in \mathsf{VP}(\mathbb{Z}_m)$.

In order to simulate the computation of a logspace-bounded oracle machine with oracle f, we will use the oracle $B = \{(x, y) : g(x, y) \neq 0\}$, which is an element of $\mathsf{laxVP}(\mathbb{Z}_m)$. If the machine we are simulating wants to obtain the value of the i-th bit of $f(z)$, we simply ask oracle B about each of the strings (x, z) for each $x \in \mathbb{Z}_m$. There will be precisely one such x for which $g(x, z)$ is equal to 0, and once we have obtained this x, we have enough information to continue the simulation of the original oracle machine, and recognize A.

The argument for $\mathsf{\Lambda P}$ is identical. (The additional characterizations for prime m are proved in [AGM15].) □

4 Strict Classes

The following lemma allows us to use ring homomorphisms to translate statements about circuits over one algebraic structure to statements about circuits over another algebraic structure.

Lemma 1. *Let R and S be commutative rings and let $\psi : R \to S$ be a function that respects addition and multiplication; i.e., for every $x, y \in R$, $\psi(x + y) = \psi(x) + \psi(y)$ and $\psi(xy) = \psi(x)\psi(y)$. Note that this is slightly weaker than the requirement that ψ be a ring homomorphism, since we do not require $\psi(1_R) = 1_S$. Let C be an arithmetic circuit consisting of $+$ and $\times$ gates, constant gates, and n input gates, and let ψC denote the same circuit with each constant $c \in R$ replaced by $\psi(c) \in S$. If the output of C on inputs $v_1, \ldots, v_n$ is w, then the output of ψC on inputs $\psi(v_1), \ldots, \psi(v_n)$ is $\psi(w)$.*

Proof. The circuit C forms a directed acyclic graph, so there exists some ordering $g_1, \ldots, g_k$ of the gates of C which forms a topological sort, i.e., such that if C contains an edge from g_i to g_j, then $i < j$. We will give an inductive proof of the stronger statement that for each gate g_i of C, if the output of g_i is w_i when the circuit is evaluated on inputs $v_1, \ldots, v_n$, then the output of the corresponding gate in ψC when ψC is evaluated on inputs $\psi(v_1), \ldots, \psi(v_n)$ is $\psi(w_i)$. The lemma is then just the special case of this statement where g_i is taken to be the output gate of C.

By our assumption on the ordering of the gates of C, g_1 has in-degree zero, so it is either a constant gate outputting $c \in R$ or an input gate x_k. If it is a constant gate, then the corresponding gate in ψC outputs $\psi(c)$ by construction; in the latter case, g_1 outputs v_k while the corresponding gate in ψC outputs $\psi(v_k)$. Thus the statement holds for $i = 1$.

Now assume that the assertion holds for gates $g_1, \ldots, g_{i-1}$, and consider g_i. If g_i is a constant gate or an input gate, the assertion holds by the same argument that we used for g_1. Suppose g_i is an addition gate $g_i = \sum_{j=1}^{l_i} g_{n_i(j)}$, with $n_i(j) < i$ for $1 \leq j \leq l_i$ by our assumption on the ordering of the gates of C. Then the output of the corresponding gate of ψC is $\sum_{j=1}^{l_i} \psi(w_{n_i(j)}) =$

$\psi\left(\sum_{j=1}^{l_i} w_{n_i(j)}\right) = \psi(w_i)$. If g_i is a multiplication gate, the assertion holds by the same argument.

□

Corollary 1. *Let R and S be commutative rings and let $\psi : R \to S$ be a ring homomorphism. (In particular, here we require $\psi(1_R) = 1_S$, in contrast to the weaker hypothesis of Lemma 1.) Then $\mathsf{strictVP}(R) \subseteq \mathsf{strictVP}(S)$ and $\mathsf{strict\Lambda P}(R) \subseteq \mathsf{strict\Lambda P}(S)$.*

Proof. We will prove the statement about $\mathsf{strictVP}$; the proof for $\mathsf{strict\Lambda P}$ is identical.

Let $A \subseteq \{0,1\}^*$ be in $\mathsf{strictVP}(R)$, and let $\{C_n \mid n \in \mathbb{N}\}$ be the corresponding arithmetic circuit family as described in Definition 1. Consider the arithmetic circuit family $\{\psi C_n \mid n \in \mathbb{N}\}$, where ψC is defined as in Lemma 1. For any string $w \in A$ of length n, the output of C_n evaluated over R on input w is 1_R, so by Lemma 1 the output of ψC_n on input w is $\psi(1_R) = 1_S$. Similarly, on any input string $w \in \{0,1\}^*$ of length n with $w \notin A$, the output of ψC_n is $\psi(0_R) = 0_R$. Thus, by the existence of the circuit family $\{\psi C_n\}$, we have $A \in \mathsf{strictVP}(S)$. □

Recall that (as shown in [AGM15]) $\mathsf{VP}(\mathbb{F}_p) = \mathsf{VP}(\mathbb{F}_{p^k})$; that is, arithmetic circuits of logarithmic depth with bounded fan-in × gates over a finite field of order equal to a power of a prime provide no more expressive power than such circuits over the finite field of the corresponding prime order. In the following theorems, we show that similar results hold in the $\mathbb{Z}_m$ setting for both $\mathsf{strictVP}$ and $\mathsf{strict\Lambda P}$. This is in some sense a neater picture than the one obtained by working over finite fields, where it is not believed in general that $\mathsf{\Lambda P}(\mathbb{F}_p) = \mathsf{\Lambda P}(\mathbb{F}_{p^k})$.

Theorem 1. *For any prime p and any integer $k \geq 1$, $\mathsf{strictVP}(\mathbb{Z}_{p^k}) = \mathsf{CC}^1[p] = \mathsf{strictVP}(\mathbb{Z}_p)$.*

Proof. The second equality was proven in [AGM15]. By Corollary 1, the containment $\mathsf{strictVP}(\mathbb{Z}_{p^k}) \subseteq \mathsf{strictVP}(\mathbb{Z}_p)$ follows from the fact that $\psi : \mathbb{Z}_{p^k} \to \mathbb{Z}_p$ given by $\psi([n]_{p^k}) = [n]_p$ is a homomorphism. It remains to prove that $\mathsf{CC}^1[p] \subseteq \mathsf{strictVP}(\mathbb{Z}_{p^k})$.

Given a $\mathsf{CC}^1[p]$ circuit C, we construct an arithmetic subcircuit over $\mathbb{Z}_{p^k}$ to simulate each gate g of C. If g is a NOT gate, $g = \neg h$, then the corresponding subcircuit is $g = (h + (p^k - 1)) \times (h + (p^k - 1))$. If g is an AND gate of fan-in two, $g = h_1 \wedge h_2$, then the corresponding subcircuit is $g = h_1 \times h_2$. Binary OR gates are handled with DeMorgan's laws.

Finally, if g is a MOD_p gate with inputs $h_1, \dots, h_l$, consider the subcircuit $g' = \left(\sum_{i=1}^{l} h_i\right)^{p^{k-1}(p-1)}$. If the original MOD_p gate evaluates to 1, then $\sum_{i=1}^{l} h_i \equiv 0 \pmod{p}$, so the sum $\sum_{i=1}^{l} h_i$ is a multiple of p. Since $p^{k-1}(p-1) \geq$

k, $\left(\sum_{i=1}^{l} h_i\right)^{p^{k-1}(p-1)}$ is a multiple of p^k, so g' evaluates to 0. If the original MOD$_p$ gate evaluates to 0, then the sum $\sum_{i=1}^{l} h_i$ is not a multiple of p and is therefore a unit in $\mathbb{Z}_{p^k}$, so (since $p^{k-1}(p-1)$ is the size of the multiplicative group $(\mathbb{Z}_{p^k})^\times$) g' evaluates to 1. Thus the subcircuit corresponding to g is simply the negation of g', using the above simulation of a NOT gate: $g = (g' + (p^k - 1)) \times (g' + (p^k - 1))$. The circuit constructed by this gate-by-gate replacement is easily seen to be a $\mathsf{strictVP}(\mathbb{Z}_{p^k})$ circuit simulating the original $\mathsf{CC}^1[p]$ circuit, so we conclude $\mathsf{CC}^1[p] \subseteq \mathsf{strictVP}(\mathbb{Z}_{p^k})$, which completes the proof. □

In order to prove the corresponding statements about the $\mathsf{strict\Lambda P}$ classes, we need the following lemma.

Lemma 2. *Let p be any prime and let k be any positive integer. If $\sigma \in (\mathbb{Z}_{p^k})^\times$ generates a subgroup of $(\mathbb{Z}_{p^k})^\times$ of order $q \in \mathsf{Supp}(p-1)$, then for any $j \in \mathbb{Z}$, either $q \mid j$ and hence $\sigma^j = 1$ or $\sigma^j - 1$ is a unit in $\mathbb{Z}_{p^k}$.*

Proof. Suppose $\sigma^j - 1$ is a non-unit in $\mathbb{Z}_{p^k}$ for some $j \in \mathbb{Z}$. We will show that $q \mid j$. Since $\sigma^j - 1$ is a non-unit, it must be that $\sigma^j - 1 = pr$ for some $r \in \mathbb{Z}$, i.e., $\sigma^j = pr + 1$. Let $\psi : \mathbb{Z}_{p^k} \to \mathbb{Z}_p$ be the ring homomorphism given by $\psi([x]_{p^k}) = [x]_p$. Then we have that $\psi(\sigma^j) = 1$, so $\psi(\sigma)^j = 1$, so the order of $\psi(\sigma)$ in $(\mathbb{Z}_p)^\times$ divides j. (This follows since ψ induces a group homomorphism from $(\mathbb{Z}_{p^k})^\times$ to $(\mathbb{Z}_p)^\times$.) But the order of $\psi(\sigma)$ divides the order of σ, which is q, so since q is prime (because $q \in \mathsf{Supp}(p-1)$), the order of $\psi(\sigma)$ is either 1 or q. If this order is q, then $j = qs$ for some $s \in \mathbb{Z}$, and thus $q|j$, which is what we needed to show.

Suppose (toward contradiction) that the order of $\psi(\sigma)$ in $(\mathbb{Z}_p)^\times$ is 1, i.e., $\psi(\sigma) = [1]_p$. The set $\psi^{-1}([1]_p)$ of congruence classes $[n]_{p^k}$ of integers $n \equiv 1 \pmod{p}$ forms a subgroup of $(\mathbb{Z}_{p^k})^\times$ of order p^{k-1}, so since (by assumption) $\sigma \in \psi^{-1}([1]_p)$, we have by Lagrange's theorem that $q \mid p^{k-1}$. Since q is prime, this implies $q = p$, which contradicts our initial assumption $q \in \mathsf{Supp}(p-1)$. □

We are now ready to prove the corresponding statement about $\mathsf{strict\Lambda P}(\mathbb{Z}_m)$ where the modulus m is a power of a prime.

Theorem 2. *For any prime p and any positive integer k, $\mathsf{strict\Lambda P}(\mathbb{Z}_{p^k}) = \mathsf{AC}^1[\mathsf{Supp}(p-1)] = \mathsf{strict\Lambda P}(\mathbb{Z}_p)$.*

Proof. As above, the second equality was proven by Allender et al. in [AGM15], and the containment $\mathsf{strict\Lambda P}(\mathbb{Z}_{p^k}) \subseteq \mathsf{strict\Lambda P}(\mathbb{Z}_p)$ is more or less immediate from Corollary 1, so it remains to show $\mathsf{AC}^1[\mathsf{Supp}(p-1)] \subseteq \mathsf{strict\Lambda P}(\mathbb{Z}_{p^k})$.

Again, we carry out a gate-by-gate simulation of an $\mathsf{AC}^1[\mathsf{Supp}(p-1)]$ circuit C, constructing an arithmetic subcircuit over $\mathbb{Z}_{p^k}$ using $+$ and $\times$ gates of the appropriate fan-in to simulate each gate g of C. A NOT gate is simulated as in the proof of Theorem 1. An AND gate $g = \wedge_{i=1}^{l} h_i$ is simulated by a single multiplication gate, $g = \prod_{i=1}^{l} h_i$. OR gates are again handled using DeMorgan's laws.

Finally, we consider the case of a MOD_q gate g with inputs $h_1, \ldots, h_l$ with $q \in \mathsf{Supp}(p-1)$. Let σ be a generator of the multiplicative subgroup of $(\mathbb{Z}_{p^k})^\times$ of order q. For each input h_i, construct a subcircuit $h_i' = 1+(\sigma-1)\times h_i$, and note that for $h_i \in \{0,1\}$, $h_i' = \sigma^{h_i}$. Construct a subcircuit $g' = \left(\left(\prod_{i=1}^l h_i'\right) - 1\right)^{p^{k-1}(p-1)}$. If $\sum_{i=1}^l h_i \equiv 0 \pmod{p^k}$, then

$$\begin{aligned}\left(\left(\prod_{i=1}^l h_i'\right) - 1\right)^{p^{k-1}(p-1)} &= \left(\sigma^{\sum_{i=1}^l h_i} - 1\right)^{p^{k-1}(p-1)} \\ &= (1-1)^{p^{k-1}(p-1)} \\ &= 0.\end{aligned}$$

If not, then $\prod_{i=1}^l h_i' = \sigma^{\sum_{i=1}^l h_i} \neq 1$, so $\sigma^{\sum_{i=1}^l h_i} - 1$ is not equal to zero and thus by Lemma 2 is a unit in $\mathbb{Z}_{p^k}$. Since $p^{k-1}(p-1) = \left|(\mathbb{Z}_{p^k})^\times\right|$, we have that $\left(\left(\prod_{i=1}^l h_i'\right) - 1\right)^{p^{k-1}(p-1)} = \left(\sigma^{\sum_{i=1}^l h_i} - 1\right)^{p^{k-1}(p-1)} = 1$. Thus we can obtain the desired value by applying a subcircuit simulating a NOT gate (as described above) to the output of g'. This gate-by-gate replacement produces a $\mathsf{strict\Lambda P}(\mathbb{Z}_{p^k})$ circuit simulating the original $\mathsf{AC}^1[\mathsf{Supp}(p-1)]$ circuit, so we conclude $\mathsf{AC}^1[\mathsf{Supp}(p-1)] \subseteq \mathsf{strict\Lambda P}(\mathbb{Z}_{p^k})$, which completes the proof. □

Again, in almost all cases we can obtain an even simpler characterization by eliminating all gates other than the MOD_p gates.

Corollary 2. *For any non-Fermat prime $p > 2$,*

$$\mathsf{strict\Lambda P}(\mathbb{Z}_{p^k}) = \mathsf{strict\Lambda P}(\mathbb{Z}_p) = \mathsf{AC}^1[\mathsf{Supp}(p-1)] = \mathsf{CC}^1[\mathsf{Supp}(p-1)].$$

Proof. We have $\mathsf{strict\Lambda P}(\mathbb{Z}_{p^k}) = \mathsf{strict\Lambda P}(\mathbb{Z}_p) = \mathsf{AC}^1[\mathsf{Supp}(p-1)]$, and it is known that for any integer m, $\mathsf{AC}^1[\mathsf{Supp}(m)] = \mathsf{AC}^1[m]$, so $\mathsf{AC}^1[\mathsf{Supp}(p-1)] = \mathsf{AC}^1[p-1]$. By a similar argument, $\mathsf{CC}^1[p-1] = \mathsf{CC}^1[\mathsf{Supp}(p-1)]$. It remains to show that $\mathsf{AC}^1[p-1] = \mathsf{CC}^1[p-1]$. Since p is an odd prime, 2 divides $p-1$, and since p is a non-Fermat prime, $p-1$ is not a power of 2. Thus $p-1$ is not a prime power, so by Theorem 10 of [AGM15], $\mathsf{AC}^1[p-1] = \mathsf{CC}^1[p-1]$. □

We now prove the following general result, which will allow us to describe $\mathsf{strictVP}(\mathbb{Z}_m)$ and $\mathsf{strict\Lambda P}(\mathbb{Z}_m)$ (for a general modulus m) in terms of the Boolean characterizations of $\mathsf{strictVP}(\mathbb{Z}_{p^k})$ and $\mathsf{strict\Lambda P}(\mathbb{Z}_{p^k})$ given above.

Theorem 3. *Let R and S be commutative rings. Then* $\mathsf{strictVP}(R \times S) = \mathsf{strictVP}(R) \cap \mathsf{strictVP}(S)$ *and* $\mathsf{strict\Lambda P}(R \times S) = \mathsf{strict\Lambda P}(R) \cap \mathsf{strict\Lambda P}(S)$.

Proof. We will prove the statement about $\mathsf{strictVP}$; the proof for $\mathsf{strict\Lambda P}$ is identical.

($\subseteq$) Since the projection functions $\psi_R : R \times S \to R$ and $\psi_S : R \times S \to S$ given by $\psi_R(r,s) = r$ and $\psi_S(r,s) = s$ are homomorphisms, the inclusion $\mathsf{strictVP}(R \times S) \subseteq \mathsf{strictVP}(R) \cap \mathsf{strictVP}(S)$ follows immediately from Corollary 1.

($\supseteq$) Let $A \subseteq \{0,1\}^*$ be in $\mathsf{strictVP}(R) \cap \mathsf{strictVP}(S)$, and let $\{C_n^R \mid n \in \mathbb{N}\}$ and $\{C_n^S \mid n \in \mathbb{N}\}$ be the associated arithmetic circuit families as described in Definition 1. Let $\chi_R : R \to R \times S$ and $\chi_S : S \to R \times S$ be given by $\chi_R(r) = (r, 0_S)$ and $\chi_S(s) = (0_R, s)$, and note that while χ_R and χ_S may not be ring homomorphisms, they do satisfy the conditions given in the statement of Lemma 1. Define a new circuit family $\{C_n \mid n \in \mathbb{N}\}$ with subcircuits $\chi_R C_n^R$ and $\chi_S C_n^S$ (in the notation of Lemma 1) so that on input w, C_n outputs $\chi_R C_n^R(w) + \chi_S C_n^S(w)$. In other words, our new circuit consists of a single binary addition gate whose inputs are the outputs of the subcircuits $\chi_R C_n^R$ and $\chi_S C_n^S$. On input $w \in \{0,1\}^*$, if $w \in A$ then the outputs of C_n^R and C_n^S are 1_R and 1_S, respectively, so by Lemma 1 the output of C_n is $\chi_R(1_R) + \chi_S(1_S) = (1_R, 0_S) + (0_R, 1_S) = 1_{R \times S}$. If $w \notin A$, then the output of C_n is $\chi_R(0_R) + \chi_S(0_S) = (0_R, 0_S) + (0_R, 0_S) = 0_{R \times S}$. Thus by the existence of $\{C_n \mid n \in \mathbb{N}\}$, we have $A \in \mathsf{strictVP}(R \times S)$. □

Note that for this result the restriction to the strict (rather than lax) versions of VP and $\mathsf{\Lambda P}$ is essential, since Corollary 1 depends crucially on the fact that circuits for $\mathsf{strictVP}$ and $\mathsf{strict\Lambda P}$ languages always output either 0 or 1. Now that we have this result, we easily obtain Boolean characterizations of $\mathsf{strictVP}(\mathbb{Z}_m)$ and $\mathsf{strict\Lambda P}(\mathbb{Z}_m)$ in the general case of a modulus m that is not a power of a prime.

Theorem 4. *For any integer $m \geq 2$, $\mathsf{strictVP}(\mathbb{Z}_m) = \bigcap_{i=1}^{l} \mathsf{CC}^1[p_i]$ and $\mathsf{strict\Lambda P}(\mathbb{Z}_m) = \bigcap_{i=1}^{l} \mathsf{AC}^1[\mathsf{Supp}(p_i - 1)]$, where $m = p_1^{k_1} \cdots p_l^{k_l}$ is the prime factorization of m.*

Proof. The characterizations of $\mathsf{strictVP}(\mathbb{Z}_m)$ and $\mathsf{strict\Lambda P}(\mathbb{Z}_m)$ as intersections of Boolean classes are simply the result of applying Theorem 3 to $\mathbb{Z}_m = \mathbb{Z}_{p_1^{k_1}} \times \cdots \times \mathbb{Z}_{p_l^{k_l}}$, using Theorem 1 and Theorem 2 for the prime power cases. □

This is perhaps not as satisfying a result as one might have hoped for. When m is a prime power, we have shown that each of $\mathsf{strictVP}(\mathbb{Z}_m)$ and $\mathsf{strict\Lambda P}(\mathbb{Z}_m)$ is equal to a single familiar Boolean class, whereas in the general case we have only been able to express the arithmetic classes as intersection of Boolean classes. As the following corollary shows, however, we can make more precise statements about $\mathsf{strict\Lambda P}(\mathbb{Z}_m)$ depending on the parity of m.

Corollary 3. *If m is even, then $\mathsf{strict\Lambda P}(\mathbb{Z}_m) = \mathsf{AC}^1$. If m is odd, then $\mathsf{AC}^1[2] \subseteq \mathsf{strict\Lambda P}(\mathbb{Z}_m)$.*

Proof. For every prime p and every integer $k \geq 1$, we have $\mathsf{AC}^1 \subseteq \mathsf{AC}^1[\mathsf{Supp}(p - 1)] = \mathsf{strict\Lambda P}(\mathbb{Z}_{p^k})$, so since $\mathsf{strict\Lambda P}(\mathbb{Z}_m)$ is the intersection of sets of the form $\mathsf{strict\Lambda P}(\mathbb{Z}_{p^k})$ we have $\mathsf{AC}^1 \subseteq \mathsf{strict\Lambda P}(\mathbb{Z}_m)$. Now assume m is even. Since 2 is one of the primes in the prime factorization of m, we have $\mathsf{strict\Lambda P}(\mathbb{Z}_m) \subseteq \mathsf{strict\Lambda P}(\mathbb{Z}_2) = \mathsf{AC}^1[\mathsf{Supp}(1)] = \mathsf{AC}^1[\varnothing] = \mathsf{AC}^1$ and so $\mathsf{strict\Lambda P}(\mathbb{Z}_m) = \mathsf{AC}^1$.

Now assume m is odd, so $m = p_1^{k_1} \cdots p_l^{k_l}$ for some odd primes $p_1, \ldots, p_l$. For each prime p_i in the factorization, $p_i - 1$ is even, so $2 \in \mathsf{Supp}(p_i - 1)$ and hence $\mathsf{AC}^1[2] \subseteq \mathsf{AC}^1[\mathsf{Supp}(p_i - 1)]$. Thus $\mathsf{AC}^1[2] \subseteq \bigcap_{i=1}^{l} \mathsf{AC}^1[\mathsf{Supp}(p_i - 1)] = \mathsf{strict\Lambda P}(\mathbb{Z}_m)$. □

Using the fact that $\mathbb{Z}$ is an initial object in the category of rings — i.e., for every ring R there exists a homomorphism $\psi : \mathbb{Z} \to R$ — we also obtain a simple Boolean characterization of $\mathsf{strict\Lambda P}(\mathbb{Z})$, as well as Boolean upper and lower bounds on $\mathsf{strictVP}(\mathbb{Z})$, and we show that $\mathsf{strictVP}(\mathbb{Z})$ and $\mathsf{strict\Lambda P}(\mathbb{Z})$ are themselves lower bounds on the classes $\mathsf{strictVP}(R)$ and $\mathsf{strict\Lambda P}(R)$, respectively, over any ring R.

Corollary 4. *Let R be any ring. Then*

- $\mathsf{strictVP}(\mathbb{Z}) \subseteq \mathsf{strictVP}(R)$ *and* $\mathsf{strict\Lambda P}(\mathbb{Z}) \subseteq \mathsf{strict\Lambda P}(R)$*,*
- $\mathsf{strict\Lambda P}(\mathbb{Z}) = \mathsf{AC}^1$*, and*
- $\mathsf{SPL} \subseteq \mathsf{strictVP}(\mathbb{Z}) \subseteq \bigcap_p \mathsf{CC}^1[p] = \bigcap_p \mathsf{VP}(\mathbb{F}_p)$*.*

Proof. As observed above, for every ring R, there exists a homomorphism $\psi : \mathbb{Z} \to R$ given by $\psi(1) = 1_R$. By Corollary 1 it follows that $\mathsf{strictVP}(Z) \subseteq \mathsf{strictVP}(R)$ and $\mathsf{strict\Lambda P}(\mathbb{Z}) \subseteq \mathsf{strict\Lambda P}(R)$.

The inclusion $\mathsf{strict\Lambda P}(\mathbb{Z}) \subseteq \mathsf{AC}^1$ follows directly by taking R to be $\mathbb{Z}_2$ and recalling that $\mathsf{strict\Lambda P}(\mathbb{Z}_2) = \mathsf{AC}^1$. We show $\mathsf{AC}^1 \subseteq \mathsf{strict\Lambda P}(\mathbb{Z})$ directly, using a gate-by-gate simulation: an OR gate is replaced by AND and NOT gates using DeMorgan's laws, a NOT gate $g = \neg h$ is simulated by a subcircuit computing $(h + (-1)) \times (h + (-1))$, and an AND gate $g = \wedge_i h_i$ is simulated by a subcircuit computing $\prod_i h_i$.

Taking $R = \mathbb{Z}_p$ and recalling that $\mathsf{strictVP}(\mathbb{Z}_p) = \mathsf{CC}^1[p]$, we obtain $\mathsf{strictVP}(\mathbb{Z}) \subseteq \mathsf{CC}^1[p]$ for every prime p and thus $\mathsf{strictVP}(\mathbb{Z}) \subseteq \bigcap_p \mathsf{CC}^1[p]$.

The complexity class SPL has been studied in [ARZ99, TH04, DKR10, PTV12, DKTV12, DK13]. It consists of all of those languages A for which there is a nondeterministic logspace machine with the property that, for all $x \in A$, the number of accepting paths is one more than the number of rejecting paths, and for all $x \notin A$, the number of accepting and rejecting paths is equal. (This class arises in the study of various versions of the perfect matching problem.) It is known that $\mathsf{L} \subseteq \mathsf{SPL}$, and under a plausible derandomization hypothesis $\mathsf{NL} \subseteq \mathsf{SPL}$. Since every function in $\#\mathsf{L}$ is in $\mathsf{VP}(\mathbb{Z})$, it is immediate from the definition that $\mathsf{SPL} \subseteq \mathsf{strictVP}(\mathbb{Z})$.

One could define a class analogous to SPL, based on functions in $\#\mathsf{SAC}^1$ instead of on $\#\mathsf{L}$, and a similar argument shows that this class is actually *identical* to $\mathsf{strictVP}(\mathbb{Z})$. This class contains the class that is called $\mathsf{UAuxPDA}(\log n, n^{O(1)})$ in [RA00], and which is shown there to coincide with SAC^1 under the same derandomization hypothesis mentioned above.

□

It is worth emphasizing that the first two parts of Corollary 4, taken together, show that for any ring R, $\mathsf{AC}^1 \subseteq \mathsf{strict\Lambda P}(R)$; in other words, polynomial size, logarithmic depth arithmetic circuits with bounded fan-in + gates and unbounded fan-in × gates are at least as powerful as AC^1 regardless of the ring over which they are evaluated, even when the outputs of the circuits are restricted to be 0 or 1. Similarly, the first and third parts of the corollary show that for any ring R, $\mathsf{SPL} \subseteq \mathsf{strictVP}(R)$.

We should also mention that no inclusion is known (in either direction) between AC^1 and SPL, or between AC^1 and $\mathsf{CC}^1[p]$ for any prime power p. (In contrast, if m is *not* a prime power, then $\mathsf{AC}^1 \subseteq \mathsf{CC}^1[m] = \mathsf{AC}^1[m]$ [AGM15].)

5 An improved complexity bound

It is mentioned in [AGM15] that all of the functions in $\Lambda\mathsf{P}(\mathbb{Z}_m)$ can be computed in ACC^1, and thus these functions lie in $\mathsf{AC}[m']$ for some m', but no attempt was made in [AGM15] to be precise about the value of m'. In this section, we make use of standard tools from number theory to pin down the correct value of m'.

For an arbitrary $x \in \mathbb{Z}_m$, consider the sequence $(x, x^2, x^3, \ldots)$. Since there are only finitely many values that each element of the sequence may take on, it is obvious that the sequence must eventually repeat a value; it is also obvious that once it does so, it will continue to repeat cyclically, since $x^{r+s} = x^s$ implies $x^{r+s+1} = x^{r+1}$, $x^{r+s+2} = x^{r+2}$, etc. We are interested in the possible values of r and s, in the notation of the previous sentence; in particular, we want to find an upper bound for r and we want to characterize the possible prime factors of s. We first describe the case in which m is a prime power.

Lemma 3. *Let p be any prime and k be any positive integer, and let $m = p^k$. Then for any $x \in \mathbb{Z}_m$, either $p \mid x$ and $x^i \equiv 0 \pmod{m}$ for all $i \geq k$ or $p \nmid x$ and there exists some minimal $l \mid (p^{k-1}(p-1))$ such that $x^l \equiv 1 \pmod{m}$.*

Note that in the notation of the above paragraph, this means that if $p \mid x$ we have $r \leq k$ and $s = 0$; otherwise, we have $r = 0$ and $s \mid (p^{k-1}(p-1))$.

Proof. Take $x \in \mathbb{Z}_m$. If $p \mid x$, then $p^k \mid x^k$, so $x^k \equiv 0 \pmod{m}$. Thus for any $i \geq k$, $x^i = x^k x^{i-k}$ is divisible by p^k, i.e., $x^i \equiv 0 \pmod{m}$.

If $p \nmid x$, then $\gcd(p, x) = 1$ (since p is prime) so $\gcd(p^k, x) = 1$ and thus x is in $(\mathbb{Z}_m)^\times$, the multiplicative group consisting of the units of $\mathbb{Z}_m$. Let l be the order of x as an element of $(\mathbb{Z}_m)^\times$. Since the order of an element of a finite group must divide the size of the group, we have that l divides $|(\mathbb{Z}_m)^\times| = \varphi(m) = p^{k-1}(p-1)$. By the definition of the order of a group element, l is minimal: for any smaller (positive) integer l', $x^{l'} \neq 1$ in $(\mathbb{Z}_m)^\times$, i.e., $x^{l'} \not\equiv 1 \pmod{m}$. □

Theorem 5. *Fix an integer $m \geq 2$, and let $m = p_1^{k_1} \cdots p_n^{k_n}$ (p_i prime, $k_i \in \mathbb{N}$) be the prime factorization of m. Given $x \in \mathbb{Z}_m$, without loss of generality let $p_1, \ldots, p_j$ be the prime factors of m that do not divide x, and let $p_{j+1}, \ldots, p_n$ be those prime factors which do divide x. Then there exist integers r and s such that $x^{r+s} = x^r$ and $x^{r+s'} \neq x^r$ for $0 < s' < s$ (i.e., the sequence $(x, x^2, x^3 \ldots)$ enters a cycle of length s starting at x^r), where $r \leq \max(k_{j+1}, \ldots, k_n)$ and $s = \operatorname{lcm}(l_1, \ldots, l_j)$ for some integers $l_1, \ldots, l_j$ such that l_i divides $p_i^{k_i-1}(p_i - 1)$.*

Proof. Note that $\mathbb{Z}_m$ is isomorphic to the product $\mathbb{Z}_{p_1^{k_1}} \times \cdots \times \mathbb{Z}_{p_n^{k_n}}$. Let $\psi : \mathbb{Z}_m \to \mathbb{Z}_{p_1^{k_1}} \times \cdots \times \mathbb{Z}_{p_n^{k_n}}$ be the natural isomorphism given by $\psi(x) = (x_1, \ldots, x_n)$ where $x_i \equiv x \pmod{p_i^{k_i}}$. We first consider the periodic behavior of each x_i in the corresponding $\mathbb{Z}_{p_i^{k_i}}$.

Note that since x and x_i differ by a multiple of p_i (in particular, a multiple of $p_i^{k_i}$), a prime factor p_i of m divides x if and only if it divides x_i. Thus for $1 \le i \le j$, we have $p_i \nmid x_i$, so Lemma 3 guarantees the existence of some $l_i \mid (p_i^{k_i}(p_i - 1))$ such that $x_i^{l_i} \equiv 1 \pmod{p_i^{k_i}}$. Let $s = \operatorname{lcm}(l_1, \ldots, l_k)$, and let $r = \max(k_{j+1}, \cdots, k_n)$.[1]

For $1 \le i \le j$, we have $l_i \mid s$. Writing $s = bl_i$, and applying the fact that $x_i^{l_i} \equiv 1 \pmod{p_i^{k_i}}$, we can see that $x_i^{r+s} = x_i^r x_i^s = x_i^r (x_i^{l_i})^b \equiv x_i^r \pmod{p_i^{k_i}}$. For $j < i \le n$, we have $k_i \le r$ and thus $k_i \le r + s$, so $x_i^{r+s} \equiv 0 \equiv x_i^r \pmod{p_i^{k_i}}$. Since x_i^{r+s} and x_i^r are congruent modulo $p_i^{k_i}$ for $1 \le i \le n$, and since x_i and x are congruent modulo $p_i^{k_i}$ as well, we have that $x^{r+s} \equiv x^r \pmod{p_i^{k_i}}$ for each i. This implies that x^{r+s} and x^r are congruent modulo m and thus are identical as elements of $\mathbb{Z}_m$. (Alternatively, the congruences $x_i^{r+s} \equiv x_i^r \pmod{p_i^{k_i}}$ tells us that $\psi(x^{r+s}) = \psi(x^r)$, and we conclude $x^{r+s} = x^r$ by the injectivity of ψ.)

It remains to show that s is minimal. Fix any integer s' with $0 < s' < s$, and suppose toward contradiction that $x^{r+s'} = x^r$. Since $s' < s$, it must be (by the definition of lcm) that $l_i \nmid s'$ for some $i \in \{1, \ldots, j\}$. Using the division algorithm, we can write $s' = al_i + b$ for some $a, b \in \mathbb{Z}$ with $0 < b < l_i$. Reducing the equality $x^{r+s'} = x^r$ modulo $p_i^{k_i}$, substituting $al_i + b$ for s', and applying the fact that $x_i^{l_i} \equiv 1 \pmod{p_i^{k_i}}$, we obtain that $x_i^r x_i^b \equiv x_i^r \pmod{p_i^{k_i}}$. Since $p_i \nmid x$ and $x \equiv x_i \pmod{p_i^{k_i}}$, we have $p_i \nmid x_i$, so x_i is a unit in $\mathbb{Z}_{p_i^{k_i}}$ and thus we can cancel it from both sides of the above congruence to obtain $x_i^b \equiv 1 \pmod{p_i^{k_i}}$. Since $0 < b < l_i$, this contradicts the minimality of l_i, and we obtain the desired contradiction.

□

The statement of Theorem 5 is more complicated than we need for our purposes, since we really only care about the question of which primes may divide s, the multiplicative period of x (i.e., the length of the cycle eventually reached by the sequence $(x, x^2, \ldots)$). Here is one way to phrase the answer to this question. As in the statement of the theorem, let $p_1, \ldots, p_j$ be the primes from the factorization of m that do not divide x. For each such p_i, let $S_i = \mathsf{Supp}(p_i - 1)$ if $k_i = 1$; otherwise, let $S_i = \mathsf{Supp}(p_i - 1) \cup \{p\}$. Then the prime factors of s all lie in $S_1 \cup \cdots \cup S_j = \left(\bigcup_{i=1}^{j} \mathsf{Supp}(p_i - 1)\right) \cup \{p_i \mid 1 \le i \le j, k_i \ge 2\}$. More generally, without confining our attention to a particular element $x \in \mathbb{Z}_m$, we can observe that the prime factors of the multiplicative period of any element of $\mathbb{Z}_m$ must

[1] The choice to let $r = \max(k_{j+1}, \cdots, k_n)$ may seem odd in light of the weaker restriction $r \le \max(k_{j+1}, \cdots, k_n)$ in the statement of the theorem. In fact, the initial part of the sequence $(x, x^2, \ldots)$ before the cycle begins may have length less than $\max(k_{j+1}, \cdots, k_n)$, but choosing r to have exactly that value works fine for our purposes.

lie in $\bigcup_{i=1}^{n} S_i$ where, by extension of the above definition, $S_i = \mathsf{Supp}(p_i - 1)$ if $k_i = 1$ and $S_i = \mathsf{Supp}(p_i - 1) \cup \{p\}$ otherwise.

We are now ready to use this result to give an upper bound on $\mathsf{lax}\Lambda\mathsf{P}(\mathbb{Z}_m)$ in terms of Boolean circuit classes.

Theorem 6. *Fix an integer m, let $m = p_1^{k_1} \cdots p_n^{k_n}$ be the prime factorization of m, and for $1 \leq i \leq n$ define $S_i = \mathsf{Supp}(p_i^{k_i - 1}(p_i - 1))$, i.e.,*

$$S_i = \begin{cases} \mathsf{Supp}(p_i - 1) & \text{if } k_i = 1 \\ \mathsf{Supp}(p_i - 1) \cup \{p_i\} & \text{if } k_i > 1. \end{cases}$$

Then $\mathsf{lax}\Lambda\mathsf{P}(\mathbb{Z}_m) \subseteq \mathsf{AC}^1\left[\bigcup_{i=1}^{n} S_i\right]$.

Proof. Consider a $\Lambda\mathsf{P}(\mathbb{Z}_m)$ circuit C. We will create a circuit C' that has subcircuits computing the Boolean truth values $[g = a]$ for each gate g of C and each $a \in \mathbb{Z}_m$. If g is the output gate of C, then $[g = 1]$ is the output gate of C'. Since the input gates of C take on only binary values (by the definition of $\Lambda\mathsf{P}(\mathbb{Z}_m)$), for each input gate g of C the subcircuit computing $[g = 1]$ is simply g and the subcircuit computing $[g = 0]$ is just $\neg g$.

If g is a $+$ gate of C (necessarily of fan-in 2), then we can simulate any gate $[g = a]$ by simply constructing a truth table using NC^0 circuitry and the $O(1)$ Boolean gates of the form $[g' = a']$, where g' is one of the two inputs to g.

Now suppose g is an unbounded fan-in × gate of C, $g = \prod_i h_i$. For each input h_i, we have a subcircuit computing the Boolean value $[h_i = a]$ for each $a \in \mathbb{Z}_m$. For each $a \in \mathbb{Z}_m$, let $v_a = |\{i \mid h_i = a\}|$, the number of inputs which are set to the value a. It is clear that $\prod_i h_i = \prod_{a \in \mathbb{Z}_m} a^{v_a}$. We will create subcircuits to compute the truth value $[a^{v_a} = b]$ for every pair of values $a, b \in \mathbb{Z}_m$. From these values, we can compute each desired output $[g = c]$ $(c \in \mathbb{Z}_m)$ by constructing a truth table using NC^0 circuitry and the $O(1)$ Boolean gates of the form $[a^{v_a} = b]$ for $a, b \in \mathbb{Z}_m$.

Fix $a \in \mathbb{Z}_m$. By Theorem 5, there exist $r, s \in \mathbb{N}$ with $r \leq \max(k_1, \ldots, k_l)$ and $\mathsf{Supp}(s) \subseteq S_1 \cup \cdots \cup S_j$ such that $a^{r+s} = a^r$. Pick $t \in \mathbb{N}$ such that $s^t \geq max(k_1, \cdots, k_l)$; then it follows that $a^{(s^t + s)} = a^{(s^t)}$ as well. Using AC^0 circuitry, we can simulate a single threshold gate outputting 1 if and only if more than s^t of the $[h_i = a]$ values are 1 (since $s^t = O(1)$). If this threshold gate outputs 1, then more than s^t of the inputs h_i to g carry the value a, so $v_a > s^t$. Thus the product $\prod_{h_i = a} h_i = a^{v_a}$ is in the cyclical part of the sequence $(a, a^2, a^3, \ldots)$ and thus is determined by the value of v_a modulo s. If this threshold gate outputs 0, then $v_a < s^t$ and so there are only $O(1)$ possibilities for a^{v_a}, each of which we can compute explicitly. In each case, the number of values to be multiplied is $O(1)$, since the values of r and s do not depend on the number of inputs to C. In the final Boolean subcircuit simulating the gate $g = \prod_i h_i$, we have the following gates.

- For $0 \leq j \leq s^t$ we have a MOD_{s^t} gate taking in the inputs $[h_i = a]$ as well as $s^t - j$ constant inputs set to 1. This outputs a 1 if and only if $v_a \equiv j$

(mod s^t). An $\wedge$ gate taking as inputs this MOD_{s^t} gate and the negation of the output of the threshold gate outputs a 1 if and only if $v_a = j$. If this $\wedge$ gate outputs a 1, then the partial product $\prod_{h_i=a} h_i = a^{v_a}$ is equal to a^j.

- For $0 \leq j < s$ we have a MOD_s gate taking in the inputs $[h_i = a]$ as well as a number of constant 1 inputs congruent to $-j$ modulo s. This outputs a 1 if and only if $v_a \equiv j \pmod{s}$. An $\wedge$ gate taking as inputs this MOD_s gate and the output of the threshold gate outputs a 1 if and only if $v_a > s^t$ and $v_a \equiv j \pmod{s}$. If this $\wedge$ gate outputs a 1, then the partial product is equal to a^{s^t+j}.

For every $a \in \mathbb{Z}_m$, we can compute the truth value $[a^{v_a} = b]$ using the outputs of the AND gates described above. Then the full product $g = \prod_i h_i$ can be computed as a product of the $O(1)$ values a^{v_a} (for $a \in \mathbb{Z}_m$) using NC^0 circuitry and the truth values $[a^{v_a} = b]$. This completes the gate-by-gate simulation. □

6 Conclusions

This paper grew out of a desire to understand Conjecture 1, which concerns the power of VP over $\mathbb{Z}_m$ for various m. It became clear that there was a need to specify more precisely what was meant by "the class of languages in $\mathsf{VP}(\mathbb{Z}_m)$ and $\mathsf{\Lambda P}(\mathbb{Z}_m)$", which led to the definition of the "strict" and "lax" classes defined and studied here. We have shown that the "strict" classes over $\mathbb{Z}_m$ correspond *precisely* to the *intersection* of the corresponding classes over $\mathbb{F}_p$ over all p dividing m. In contrast, the "lax" classes provide enough computational power to simulate the *union* of these same classes. It would be interesting to know more about various inclusions, such as $\mathsf{VP}(\mathbb{Z}_2) \cup \mathsf{VP}(\mathbb{Z}_3) \subseteq \mathsf{L}^{\mathsf{VP}(\mathbb{Z}_6)} \subseteq \mathsf{AC}^1[6]$. Is there any reason to think that either the upper bound or the lower bound on the complexity of $\mathsf{VP}(\mathbb{Z}_6)$ might be more-or-less tight?

Acknowledgments

The first author acknowledges the support of NSF grant CCF-1555409. This work was carried out while the second author was a participant in the 2015 DIMACS REU program at Rutgers University, supported in part by NSF grant CCF-1263082.

References

[AGM15] Eric Allender, Anna Gál, and Ian Mertz. Dual VP classes. In *Symposium on Mathematical Foundations of Computer Science (MFCS)*, LNCS. Springer, 2015. to appear.

[AJMV98] Eric. Allender, Jia Jiao, Meena Mahajan, and V. Vinay. Noncommutative arithmetic circuits: Depth reduction and size lower bounds. *Theoretical Computer Science*, 209:47–86, 1998.

[ARZ99] Eric Allender, Klaus Reinhardt, and Shiyu Zhou. Isolation, matching, and counting: Uniform and nonuniform upper bounds. *Journal of Computer and System Sciences*, 59(2):164–181, 1999.

[BBR94] David A. Mix Barrington, Richard Beigel, and Steven Rudich. Representing boolean functions as polynomials modulo composite numbers. *Computational Complexity*, 4:367–382, 1994.

[Bür99] Peter Bürgisser. On the structure of Valiant's complexity classes. *Discrete Mathematics & Theoretical Computer Science*, 3(3):73–94, 1999.

[Bür00] Peter Bürgisser. Cook's versus Valiant's hypothesis. *Theoretical Computer Science*, 235(1):71–88, 2000.

[DK13] Samir Datta and Raghav Kulkarni. Space complexity: What makes planar graphs special? *Bulletin of the EATCS*, 109:35–53, 2013.

[DKR10] Samir Datta, Raghav Kulkarni, and Sambuddha Roy. Deterministically isolating a perfect matching in bipartite planar graphs. *Theory Comput. Syst.*, 47(3):737–757, 2010.

[DKTV12] Samir Datta, Raghav Kulkarni, Raghunath Tewari, and N. V. Vinodchandran. Space complexity of perfect matching in bounded genus bipartite graphs. *Journal of Computer and System Sciences*, 78(3):765–779, 2012.

[GW96] Anna Gál and Avi Wigderson. Boolean complexity classes vs. their arithmetic analogs. *Random Struct. Algorithms*, 9(1-2):99–111, 1996.

[KP11] Pascal Koiran and Sylvain Perifel. Interpolation in Valiant's theory. *Computational Complexity*, 20(1):1–20, 2011.

[LL76] Richard E. Ladner and Nancy A. Lynch. Relativization of questions about log space computability. *Mathematical Systems Theory*, 10:19–32, 1976.

[PTV12] Aduri Pavan, Raghunath Tewari, and N. V. Vinodchandran. On the power of unambiguity in log-space. *Computational Complexity*, 21(4):643–670, 2012.

[RA00] Klaus Reinhardt and Eric Allender. Making nondeterminism unambiguous. *SIAM Journal on Computing*, 29:1118–1131, 2000.

[TH04] Thomas Thierauf and Thanh Minh Hoang. On closure properties of GapL. *Electronic Colloquium on Computational Complexity (ECCC)*, (024), 2004.

[Val79] Leslie G. Valiant. Completeness classes in algebra. In *Proc. 11th ACM STOC*, pages 249–261, 1979.

[VSBR83] Leslie G. Valiant, S. Skyum, S. Berkowitz, and C. Rackoff. Fast parallel computation of polynomials using few processors. *SIAM Journal on Computing*, 12(4):641–644, 1983.

[Wil14] Ryan Williams. The polynomial method in circuit complexity applied to algorithm design (invited talk). In *Conference on Foundations of Software Technology and Theoretical Computer Science (FST&TCS)*, number 29 in LIPIcs, pages 47–60. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2014.