A Note on *Mod* and Generalised *Mod* Classes

Meena Mahajan* N. V. Vinodchandran†

Keywords. Computational Complexity: Mod classes, relativised separations, truth-table reductions.

1 Introduction

We characterise *Mod* classes in terms of $\#P$ functions, where the membership is determined by co-primality or gcd testing of the function value (Theorem 3.1), instead of residue $\pmod{k}$ testing. Imposing a restriction on the range of the functions gives a characterisation of the intersection of *Mod* classes (Theorem 3.2). These intersection classes, which we denote by $Mod\cap_k P$, are interesting because they share most of the "nice" properties (closure under complementation, normal forms, lowness for itself etc) of Mod_pP for prime p. We show that the class $Mod\cap_k P$ is low for Mod_kP, and also for $Mod\cap_k P$ itself (Theorem 3.3).

We also strengthen some of the separation results known for *Mod* classes. A diagonalisation argument due to Beigel shows that when k is a prime not dividing j, Mod_jP can be separated from Mod_kP in some relativised world. We observe that this argument even separates $Mod\cap_j P$ from Mod_kP under the same conditions (Theorem 4.1). Further, if k is not known to be prime, the same argument still diagonalises, but out of a smaller class; it separates $Mod\cap_j P$ from $Mod\cap_k P$ (Theorem 4.2).

The class $ModP$ was defined in [6] as a generalisation of the *Mod* classes. We define a simple generalisation, $ModKP$, and show that it coincides with the disjunctive truth table closure of $ModP$, P_{dtt}^{ModP} (Theorem 5.2). We give neat characterisations of P_{dtt}^{ModP} and P_{ctt}^{ModP} (Theorem 5.3), and also a new characterisation of $ModP$ (Theorem 5.4).

The results of section 5 thus give us an overall picture of the relations between the generalised *Mod* classes as shown in Figure 1. Arrows denote containment, and connections tagged *co-* indicate that the corresponding classes are the Co-classes of each other.

*The Institute of Mathematical Sciences, Madras 600 113, India. email: meena@imsc.ernet.in

†The Institute of Mathematical Sciences, Madras 600 113, India. email: vinod@imsc.ernet.in. Work done at the Department of Computer Science and Engineering, Indian Institute of Technology, Madras 600 036, India.

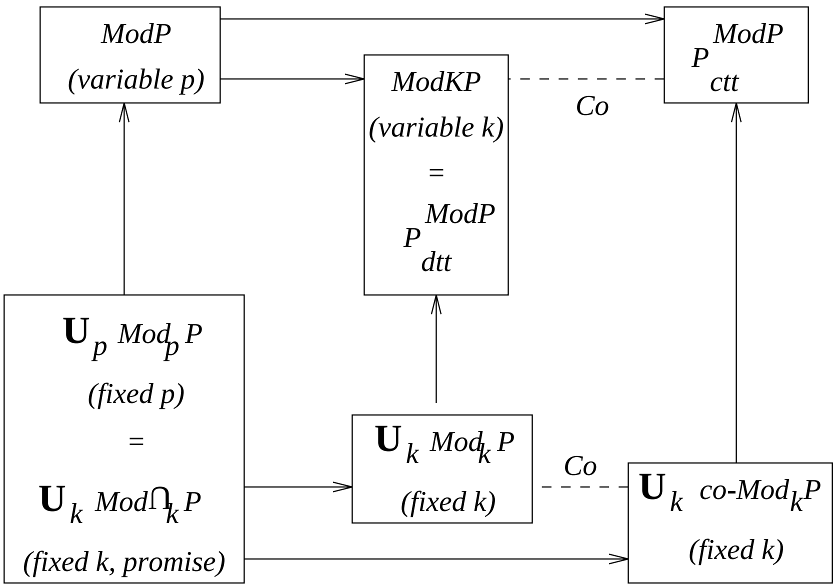

Figure 1. Relations among gemeralised Mod classes

2 Preliminaries

We follow the standard definitions and notations in computational complexity theory (see, e.g., [1] or [5]). A function f is in $\#P$ if there exists a nondeterministic polynomial time Turing Machine M such that $\forall x$, $f(x)$ is the number of accepting paths of M on input x. $\#P$ is closed under several arithmetic operations, including addition, multiplication, binomial coefficients, etc. [3]. A language L is in Mod_kP [3, 4] if there is a $\#P$ function f such that $\forall x \in \Sigma^*$, $x \in L \Leftrightarrow f(x) \not\equiv 0 \pmod k$. We use the notation $\vee_i Mod_{p_i}P$ and $\wedge_i Mod_{p_i}P$ to denote the classes $\{\cup_i L_i \mid L_i \in Mod_{p_i}P\}$ and $\{\cap_i L_i \mid L_i \in Mod_{p_i}P\}$ respectively. If k has prime factorisation $k = \Pi_i p_i^{\alpha_i}$ where the p_i are distinct primes, then we denote $\Pi_i p_i$ by $\pi(k)$. If $\pi(k) = k$ then k is said to be squarefree. $\phi(k)$ denotes the Eulerian function, the number of integers less than and co-prime to k.

The following results about *Mod* classes will be used in this note.

Theorem 2.1 *1.* ([3], Corollary 33) $Mod_kP = \vee_{p|k,\ p\ prime} Mod_pP$.

2. ([3], Theorem 23) *Let p be prime. A language L is in Mod_pP if and only if it has a 0-1 normal form $\#P$ function; that is, there exists a $\#P$ function satisfying, for all x,*

$$\begin{aligned} x \in L &\Rightarrow f(x) \equiv 1 \pmod p \\ x \notin L &\Rightarrow f(x) \equiv 0 \pmod p \end{aligned}$$

3. ([3], Theorem 27) *If p is prime, then $Mod_pP^{Mod_pP} = Mod_pP$.*

4. ([2], Theorem 10) *Let $j > 1$, and let k be a prime number that is not a divisor of j. There exists an oracle A such that $Mod_jP^A \not\subseteq Mod_kP^A$.*

The class $ModP$ has been introduced by Köbler et al in [6]. $ModP$ is the generalised version of Mod_pP, where p is a prime.

Definition 2.2 ([6]) *A language L is in $ModP$ iff there exists a $\#P$ function f and a function $g \in FP$ such that for all strings x, $g(x) = 0^p$ for some prime p, and $x \in L \Leftrightarrow f(x) \not\equiv 0 \pmod p$.*

It is shown in [6] that the $\#P$ function f can be brought into 0-1 normal form (it always evaluates to either 1 or 0 $\pmod{|g(x)|}$). It has also been shown that the class does not change if, in the definition, g is allowed to return powers of prime numbers.

3 New Characterisations

In this section, we show some new characterisations for *Mod* classes. These characterisations use co-primality and gcd testing on the values of #P functions, rather than residue testing.

Theorem 3.1 *$L \in Mod_k P$ if and only if there exists a $\#P$ function f such that $x \in L \Leftrightarrow$ $gcd(f(x), k) \neq 1$.*

Proof: $(\Rightarrow)$. Let $L \in Mod_k P$. Let $k = \Pi_i p_i^{\alpha_i}$, where p_i are the prime factors of k. Then

$$
\begin{array}{lll}
& L \in \vee_{p_i|k,\ p_i\ prime} Mod_{p_i} P & \text{Theorem 2.1, 1.} \\
\Rightarrow & \overline{L} \in \wedge_{p_i|k,\ p_i\ prime} \text{ Co-}Mod_{p_i} P & \\
\Rightarrow & \overline{L} \in \wedge_{p_i|k,\ p_i\ prime} Mod_{p_i} P &
\end{array}
$$

So let $\overline{L} = \cap L_i$ where each $L_i \in Mod_{p_i} P$ via $\#P$ function f_i in 0-1 normal form (from Theorem 2.1, 2). Then it is easy to verify that the function $f = \Sigma_i \frac{k}{p_i^{\alpha_i}} f_i$ satisfies the given conditions.

$(\Leftarrow)$. If f is a $\#P$ function satisfing the given conditions then $L \in Mod_k P$ via the $\#P$ function $h = f^{\phi(k)} + (k-1)$. ∎

In other words, a language $L \in Mod_k P$ can be characterised using a $\#P$ function f such that if $x \in L$, f maps x to a non-invertible element of the ring $\mathbf{Z}/k\mathbf{Z}$, and if $x \notin L$, f maps x to an invertible element.

A promise version of the above class of functions, where

$$gcd(f(x), k) \neq 1 \Rightarrow gcd(f(x), k) = k$$

characterises the intersection of the $Mod_{p_i} P$ classes, where p_i is a prime factor of k. For brevity we henceforth denote this class by

$$Mod \cap_k P \triangleq \cap_{p_i|k,\ p_i\ prime} Mod_{p_i} P$$

Thus if k is squarefree, then $Mod \cap_j P = Mod \cap_k P$ for all j such that $\pi(j) = k$.

Theorem 3.2 *$L \in Mod \cap_k P$ iff there exists a $\#P$ function f such that*

$$
\begin{array}{lll}
x \in L & \Rightarrow & gcd(f(x), k) = 1 \\
x \notin L & \Rightarrow & gcd(f(x), k) = k
\end{array}
$$

Proof: $(\Rightarrow)$. Let $L \in Mod \cap_k P$ where $k = \Pi_i p_i^{\alpha_i}$. For all i, let $L \in Mod_{p_i} P$ via $\#P$ functions h_i in 0-1 normal form. Then $L \in Mod_{p_i^{\alpha_i}} P$ via $\#P$ function $f_i = h_i^{\alpha_i \phi(p_i^{\alpha_i})}$ which is also in 0-1 normal form. Now it is easy to verify that the function $f = \Sigma_i \frac{k}{p_i^{\alpha_i}} f_i$ satisfies the given conditions.

($\Leftarrow$). It is obvious that if there exists a function satisfying the conditions, then $L \in Mod_{p_i}P$ for each i via the same function. Hence $L \in Mod\cap_k P$. ∎

It follows that for every k, languages in $Mod\cap_k P$ have a 0-1 normal form #P function with respect to k. Note that in the above theorem, the conditions of Theorem 3.1 have been restricted to the promise version *and* inverted. This does not matter because $Mod\cap_k P$ is closed under complementation.

The class $Mod\cap_k P$ is of some interest because it is low for Mod_kP, as we show below. In fact, it is also low for itself, whereas an analogous result for Mod_kP classes is known to hold only when k is prime. Also, we do not know of any class which contains $Mod\cap_k P$ *and* is low for Mod_kP; $Mod\cap_k P$ is the largest known class with this property.

Theorem 3.3 *For any $k \geq 2$,*

(1) $Mod_kP^{Mod\cap_k P} = Mod_kP$

(2) $Mod\cap_k P^{Mod\cap_k P} = Mod\cap_k P$

Proof: We prove (1); (2) follows identically. Let $A \in Mod_kP^{Mod\cap_k P}$ via an oracle $L \in Mod\cap_k P$. Then

$$\begin{array}{rcll} A & \in & \vee_{p_i|k,\ p_i\ prime} Mod_{p_i}P^L & \text{relativised version of Theorem 2.1, 1.} \\ & \subseteq & \vee_{p_i|k,\ p_i\ prime} Mod_{p_i}P^{Mod_{p_i}P} & \text{by definition of } Mod\cap_k P \\ & = & \vee_{p_i|k,\ p_i\ prime} Mod_{p_i}P & \text{Theorem 2.1, 3.} \\ & = & Mod_kP & \text{Theorem 2.1, 1.} \end{array}$$

∎

4 Separation Results

In [2], the construction of an oracle relative to which Mod_jP is not contained in Mod_kP is outlined (Theorem 2.1, 4.). This result applies when k is prime and j and k are relatively prime. It is open whether the second condition alone is sufficient to exhibit such a separation. If we consider separations of the $Mod\cap$ classes instead of Mod classes, then we show (Theorem 4.2) that this condition suffices.

A careful examination of the oracle construction in [2] shows that only the subset $Mod\cap_j P$ of Mod_jP is used in proving $Mod_jP^A \not\subseteq Mod_kP^A$. The construction diagonalises out of the class Mod_kP, in the process creating a language which satisfies the promise of Theorem 3.2. Thus the construction actually proves the following result:

Theorem 4.1 *Let $j > 1$, and let k be a prime that is not a divisor of j. Then there exists an oracle A such that*

$$Mod\cap_j P^A \not\subseteq Mod_k P^A$$

If k is allowed to be composite, as long as it has at least one prime factor *not* dividing j, the diagonalisation argument can still be used. However, it now diagonalises out of a much smaller (presumably) class, namely the class $Mod\cap_k P$.

Theorem 4.2 *Let j and k be two integers. If k has a prime factor not dividing j, then there exists an oracle B such that $Mod\cap_j P^B \not\subseteq Mod\cap_k P^B$.*

Proof: Since k has a prime factor p that does not divide j, it follows from the above theorem that there exists an oracle B such that $Mod\cap_j P^B \not\subseteq Mod_p P^B$. But $Mod\cap_k P^B \subseteq Mod_p P^B$, since $p|k$. Therefore $Mod\cap_j P^B \not\subseteq Mod\cap_k P^B$. ∎

In particular, if $gcd(j,k) = 1$, then the corresponding $Mod\cap P$ classes can be separated; there exists an oracle B such that $Mod\cap_j P^B \not\subseteq Mod\cap_k P^B$.

For any two primes p, q, the classes $Mod_p P$ and $Mod_q P$ can be separated (from Theorm 2.1, 4.) in some relativised world. Consequently, we have a proper separation between the $Mod\cap_k P$ and $Mod_k P$ classes in some relativised world, as the following corollary states .

Corollary 4.3 *If k is not prime or a power of a prime, then there is an oracle C such that $Mod\cap_k P^C \subset Mod_k P^C$.*

5 Generalised Mod classes

In this section we generalise the class $Mod_k P$ to $ModKP$ and show that this class is precisely the disjunctive truth table closure of the class $ModP$.

Definition 5.1 *A language L is in $ModKP$ iff there exists a $\#P$ function f and a function $g \in FP$ such that for all strings x, $g(x)$ outputs a positive integer k as a list $\langle 0^{p_1^{\alpha_1}}, 0^{p_2^{\alpha_2}}, \cdots, 0^{p_n^{\alpha_n}} \rangle$ where $k = \Pi_i p_i^{\alpha_i}$, and $x \in L \;\Leftrightarrow\; f(x) \not\equiv 0 \pmod{k}$.*

(k can also be represented as a list $\langle\langle 0^{p_1}, 0^{\alpha_1}\rangle, \langle 0^{p_2}, 0^{\alpha_2}\rangle, \cdots, \langle 0^{p_n}, 0^{\alpha_n}\rangle\rangle$. Even though $0^{p_i^{\alpha_i}}$ requires $p_i^{\alpha_i}$ to be polynomially bounded (implying small exponents), the same number $p_i^{\alpha_i}$ with polynomial-valued α_i can be expressed simply by repeating 0^{p_i} α_i times in the list.)

Theorem 5.2 $P_{dtt}^{ModP} = ModKP$

Proof: (a) $ModKP \subseteq P_{dtt}^{ModP}$.

Let $L \in ModKP$ via $f \in \#P$ and $g \in FP$. Define $B = \{\langle x, 0^{p^e}\rangle \mid f(x) \not\equiv 0 \pmod{p^e}\}$. Then $B \in ModP$ via $\#P$ function f and FP function g_B, where g_B, on input $\langle x, 0^{p^e}\rangle$, outputs 0^{p^e}. (Although g does not return a prime, it always returns a power of a prime. So the language is still in $ModP$, as described in [6].)

Let $g(x) = \langle 0^{p_1^{\alpha_1}}, 0^{p_2^{\alpha_2}}, \cdots, 0^{p_n^{\alpha_n}}\rangle$, representing $k = \Pi_i p_i^{\alpha_i}$. Now L disjunctively reduces to B via an FP function h, where h, on input x, produces the list $\langle\langle x, 0^{p_1^{\alpha_1}}\rangle, \langle x, 0^{p_2^{\alpha_2}}\rangle, \cdots, \langle x, 0^{p_n^{\alpha_n}}\rangle\rangle$.

(b) $P_{dtt}^{ModP} \subseteq ModKP$.

Let L be disjunctively reducible to a set $B \in ModP$ via h. Then for all strings x, $h(x)$ produces a list $\langle y_1, y_2, \cdots, y_m\rangle$ such that $x \in L \Leftrightarrow \exists i, 1 \leq i \leq m : y_i \in B$.

Let $B \in ModP$ via a 0-1 normal form $\#P$ function f and an FP function g. For any string x, let $P(x) = \{|g(y_1)|, |g(y_2)| \cdots |g(y_m)|\}$ be the set of primes computed by g. (Note that two strings may give the same prime on same input x.) Let $I_p(x) = \{y_i \mid g(y_i) = 0^p\}$. Define functions $\tilde{f}$ and $\tilde{g}$ as follows:

$$\tilde{f} = \sum_{p \in P(x)} \left\{ \left(\prod_{q \in P(x)-p} q \right) \left(\left(\prod_{y \in I_p(x)} (f(y)+p-1)^{p-1} \right) (p-1)+1 \right) \right\}$$

$$\tilde{g}(x) = \langle 0^{p_1}, 0^{p_2}, \cdots, 0^{p_n}\rangle \text{ each } p_i \in P(x)$$

Since the value of each prime is polynomial in the length of x, it follows from the closure properties of $\#P$ functions that $\tilde{f} \in \#P$. Also it is easy to verify that $\tilde{g} \in FP$. We show that the language $L \in ModKP$ via $\tilde{f} \in \#P$ and $\tilde{g} \in FP$.

Let the value that $\tilde{g}$ computes on input x be k.

$$\begin{aligned}
x \in L \quad &\Rightarrow \quad f(y_i) \equiv 1 \pmod{|g(y_i)|} \text{ for some } i \leq m \\
&\Rightarrow \quad f(y_i) \equiv 1 \pmod{p} \text{ for some } p \in P(x),\ p = |g(y_i)| \\
&\Rightarrow \quad \left(\textstyle\prod_{y \in I_p(x)} (f(y)+p-1)^{p-1}\right)(p-1)+1 \equiv 1 \pmod{p} \\
&\Rightarrow \quad \tilde{f}(x) \not\equiv 0 \pmod{p} \\
&\Rightarrow \quad \tilde{f}(x) \not\equiv 0 \pmod{k} \\
x \notin L \quad &\Rightarrow \quad f(y_i) \equiv 0 \pmod{|g(y_i)|} \text{ for all } i \leq m \\
&\Rightarrow \quad f(y_i) \equiv 0 \pmod{p}, p = |g(y_i)|\ \forall i \\
&\Rightarrow \quad \left(\textstyle\prod_{y \in I_p(x)} (f(y)+p-1)^{p-1}\right)(p-1)+1 \equiv 0 \pmod{p}\ \forall p \in P(x) \\
&\Rightarrow \quad \tilde{f}(x) \equiv 0 \pmod{k}
\end{aligned}$$

■

Like Mod_kP, $ModKP$ can also be characterised in terms of gcd testing.

Theorem 5.3 *A language $L \in ModKP$ if and only if there exists a $\#P$ function f and a function $g \in FP$ such that for all strings x, $g(x)$ outputs a positive integer k as a list $\langle 0^{p_1^{\alpha_1}}, 0^{p_2^{\alpha_2}}, \cdots, 0^{p_n^{\alpha_n}} \rangle$ where $k = \Pi_i p_i^{\alpha_i}$, and*

$$x \in L \;\Leftrightarrow\; gcd(f(x), k) \;\neq\; 1$$

Proof: (a) Let L be in $ModKP$ via functions $h \in \#P$, $g \in FP$. Then f as defined below satisfies the required condition. Let $k = \Pi_i p_i^{\alpha_i}$ be computed by g on input x. Then

$$f(x) = \Sigma_i \frac{k}{p_i^{\alpha_i}} \Big[(h(x))^{p_i - 1}(p_i - 1) + 1 \Big].$$

(b) Checking if $gcd(f(x), k) = 1$ is conjunctive truth-table reducible to $ModP$ queries (using a construction similar to that in the proof of Theorem 5.2 (a)). Since $ModP$ is closed under complementation [6], $ModKP = P_{dtt}^{ModP} =$ co-P_{ctt}^{ModP}. ■

The preceding theorem also characterises P_{ctt}^{ModP} as the class of languages L such that $x \in L \Leftrightarrow gcd(f(x), k) = 1$, where f and g are as defined in the theorem.

For $ModP$, the $\#P$ function can be brought into 0-1 normal form. If the FP function is also allowed to return non-primes in suitably encoded form, we get the presumably larger class $ModKP = P_{dtt}^{ModP}$. However, if the $\#P$ function is constrained to be in 0-1 normal form with respect to these composite numbers as well, we get back the original class $ModP$, as shown below.

Theorem 5.4 *A language $L \in ModP$ if and only if there exists a function $f \in \#P$ and a function $g \in FP$ such that for all strings x, $g(x) = 0^k$ for some positive integer $k \geq 2$ and*

$$x \in L \;\Rightarrow\; f(x) \equiv 1 \pmod{k}$$
$$x \notin L \;\Rightarrow\; f(x) \equiv 0 \pmod{k}$$

(Larger values of k can be represented using the list representation, as in the definition of $ModKP$. However, it is easy to see that in this case, this makes no difference to the class.)

Proof: ($\Rightarrow$) This follows from the 0-1 normal form of $ModP$.

($\Leftarrow$) Suppose there exist functions $f \in \#P$ and $g \in FP$. Consider the function g' which returns any prime factor of the number computed by g on x. Since g returns $g(x)$ in unary (or in factorised) notation, clearly $g' \in FP$. Now $L \in ModP$ via f and g'. ■

Acknowledgements

We wish to thank Johannes Köbler for pointing out an error in an earlier version of this note. We also wish to thank V Arvind and V Vinay for many comments which helped improve the presentation and readability of this note.

References

[1] J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity – I.* Springer Verlag, Berlin Heidelberg, 1988.

[2] R. Beigel. Relativized counting classes: relations among thresholds, parity, and mods. *Journal of Computer and System Sciences*, 42:76–96, 1991.

[3] R. Beigel, J. Gill, and U. Hertrampf. Counting classes: Thresholds, parity, mods, and fewness. In *Proceedings of the Seventh Annual Symposium on Theoretical Aspects of Computer Science*, pages 49–57. Springer-Verlag, 1990. Lecture Notes in Computer Science # 415.

[4] U. Hertrampf. Relations among Mod-classes. *Theoretical Computer Science*, 74:325–328, 1990.

[5] J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 1979.

[6] J. Köbler and S. Toda. On the power of generalized MOD-classes. In *Proceedings of the Eighth Annual Conference on Structure in Complexity theory*, 1993.