A Note on Unambiguous Function Classes

Sven Kosub

Theoretische Informatik, Julius-Maximilians-Universität Würzburg, Am Hubland, D-97074 Würzburg, Germany

Key words: Computational complexity, unambiguous computation, function classes

1 Introduction

Unambiguous computation according to UP has become a classical notion in computational complexity theory. Unambiguity is also used in a theorem of Wagner [14]. A set $L$ is in $P^{NP}$ iff there are a set $A \in NP$ and a polynomial $p$ such that for all $x$ and $y$ with $|y| \leq p(|x|)$, if $(x, y) \in A$ then $(x, y - 1) \in A$, and $x \in L$ iff the maximal $y$ with $(x, y) \in A$ is odd. Here, essentially for each $x$ all positive $y$'s must be neighbored, they form a cluster, where 0 is always in the cluster if one exists. Unambiguity is reflected in the existence of at most one such cluster. This construction is crucially relevant for parallel access to $NP$ (see [14]).

The same idea occurs in the definition of the complexity-theoretic operator F [13]: $F\mathcal{K}$ is the class of all functions $f$ for which there exist an $A \in \mathcal{K}$ and polynomial $p$ as above with $f(x) = \sup\{y||y| \leq p(|x|) \wedge (x, y) \in A\}$. This operator plays a key role as part of an operator pair to compare function classes and set classes. Note that maximization can be replaced with cardinality whenever $\mathcal{K}$ is closed under $\leq_m^P$-reductions.

Above, all clusters contain 0 as minimal element. In this note, we drop this requirement. Let us consider (polynomially time-bounded) nondeterministic machines where the set of accepting paths is always a cluster. As we will see, such cluster machines can be easily transformed into UP-machines. The point why it is nevertheless sensible to study these machines is that cluster machines capture UP-machines that actually can count. This feature is expressed in the unambiguous counting class $c\#P$--the $\#P$-version for cluster machines.

The class $c\#P$ shows astonishing behavior with respect to closure properties. A theory of closure properties for complexity classes of functions was established...
in [8]. A class $\mathcal{F}$ is closed under $f$ if the composition of $f$ with any function in $\mathcal{F}$ is again a function in $\mathcal{F}$. Of highest importance are (non-constructively) hard closure properties—a function $f$ is hard for $\mathcal{F}$ if the case that $\mathcal{F}$ is closed under $f$ is equivalent to $\mathcal{F}$ being closed under every function from $\mathcal{F}$. Using the structural assumption $PP = UP$, it was proved that, e.g., division is a hard closure property for $\#P$ [8].

First, we prove that $c\#P$ also possesses hard closure properties. This is surprising insofar as $c\#P$ is a promise class, i.e., a class without an obvious enumeration of its underlying machines. Even so, we identify operations (expressed by $c\#P$-functions) being able to simulate any other operation (expressible as a $c\#P$-function) over cluster machines.

A second remarkable property is the following: It was observed that $\#P$ has intermediate closure properties (e.g., minimization [8]), i.e., functions that are neither hard nor is $\#P$ closed under them. Provably, $c\#P$ does not have (unary) intermediate closure properties. We give a criterion for deciding if a function is $c\#P$-hard or $c\#P$ is closed under it.

From [8,2,3], hard closure properties are expected to reduce the number of witnesses, e.g., division for $\#P$. Here, however, the situation is completely different. We show that incrementation is $c\#P$-hard, while $c\#P$ is closed under division. Hence, this is an instructive example where the term “witness reduction” is not to be taken literally, but rather in the sense of simplifying the witness set (see the discussion in [2, pp. 412-415]).

Unambiguous clustering is furthermore studied in a more general setting for counting and optimization. Remember that for “fixed” clusters (as in the definition of the operator $F$), maximization and counting coincide. For “free” clusters, this is not to be expected. We consider three unambiguous operators corresponding to those in [11,4]: counting ($c\#$), maximization ($c\text{max}$), and minimization ($c\text{min}$) within clusters. These operators are compared to each other—giving a taxonomy of classes similar to [4]—and to the related ambiguous operators in the context of classes of the polynomial hierarchy. Mostly, proofs are straightforward, but they show in an intuitive way how strong clustering and unambiguity are connected.

2 Preliminaries

We use the finite alphabet $\Sigma = \{0, 1\}$. The cardinality of a set $L \subseteq \Sigma^*$ is denoted by $\#L$. Let $c_L$ denote the characteristic function of $L$. Usually we interpret a word $x$ as the binary encoding of a natural number. For closure properties, we need a bijection between $\Sigma^*$ and $N$. We suppose the standard
lexicographical ordering of $\Sigma^*$. For a function $f : \Sigma^* \to \Sigma^*$, the notation $\lambda x. f(x)$ stands for the mapping $x \mapsto f(x)$. The function $\text{sgn} : \mathbb{N} \to \mathbb{N}$ is defined as $\text{sgn} x = 1$, if $x > 0$, and $\text{sgn} x = 0$, if $x = 0$.

Our computational model is the standard nondeterministic Turing machine [9]. $M(x)$ denotes the computation tree of a machine $M$ on input $x$. In case of polynomial time resources we suppose complete binary computation trees. For computation trees $M(x)$, we define $\text{acc}_M(x)$ as the set of all accepting paths of $M(x)$, and $\text{out}^+_M(x)$ as the set of all outputs on accepting (rejecting, resp.) paths of $M(x)$. We use some shorthands for machine types: NTM stands for nondeterministic Turing machine, and NPTM for polynomial-time NTM.

We deal with the standard complexity classes $\text{P}, \text{UP}, \text{coUP}, \text{NP}, \text{coNP}, \text{PP}$, $\text{C}_\text{P}, \text{PH}, \text{CH}$, and their relativizations [5,9]. To describe function classes, we use the modern operator notation provided in [11,13,4]. A function $f$ is in $\# \cdot \mathcal{K}$ if there exist a set $A \in \mathcal{K}$ and a polynomial $p$ such that $f(x) = \# \{y | y = p(|x|) \}$ and $(x,y) \in A$ for every $x$ [11]. We abbreviate such witness sets as $A^p_\mathcal{K}$. A function $f$ is in $\text{max} \cdot \mathcal{K}$ if there is an $A \in \mathcal{K}$ and a polynomial $p$ such that $f(x) = \text{max} A^p_\mathcal{K}$ for every $x$ [4]. Replacing max by min gives the class $\text{min} \cdot \mathcal{K}$. To consider total functions we define maximum and minimum of the empty set to be 0, what is in the second case a little different from the original definition [4] but is more compatible with the partial case. $\text{FP} = \text{F} \cdot \text{P}$ [13] denotes the class of functions computable in deterministic polynomial time.

3 Clusters, Unambiguous Counting, and Closure Properties

We formalize the concept of clusters by introducing certain equivalence relations. Let $M$ be an NTM, and let $y$ and $z$ be paths of $M(x)$ for input $x \in \Sigma^*$. We write $y \equiv_M(x) z$ iff

- $|y - z| = 1$ (i.e., $y$ and $z$ are neighboring paths) and
- $M(x)$ accepts on path $y$ if and only if $M(x)$ accepts on path $z$.

The relation $\equiv^*_{M(x)}$ (the reflexive and transitive closure of $\equiv_{M(x)}$) is an equivalence relation partitioning $M(x)$ in equivalence classes. A cluster is an equivalence class whose representatives are accepting paths of $M(x)$.

**Definition 1** An NTM $M$ is a cluster machine if and only if for every $x \in \Sigma^*$, there is a path $y$ of $M(x)$ such that $\{ z | z \equiv^*_{M(x)} y \land y \in \text{acc}_M(x) \} = \text{acc}_M(x)$.

An NTM $M$ computes a function $f$ uniquely iff $\text{out}^+_M(x) = \{ f(x) \}$ and $\# \text{acc}_M(x) = 1$ for every $x$, and $M$ computes $f$ almost-uniquely iff $f(x) > 0$ implies $\text{out}^+_M(x) = \{ f(x) \} \land \# \text{acc}_M(x) = 1$ and $f(x) = 0$ implies $\text{out}^+_M(x) = \emptyset$, both...
for every $x$.

**Theorem 2** (1) If an NPTM $M$ computes $f$ uniquely, then there is a polynomial-time cluster machine (PCM) $N$ such that $\#	ext{acc}_N(x) = f(x)$ for all $x \in \Sigma^*$.

(2) For every PCM $N$ fulfilling $\#	ext{acc}_N(x) > 0$ for every $x \in \Sigma^*$, there is an NPTM $M$ that computes the function $\#	ext{acc}_N$ uniquely.

(3) There is an NPTM $M$ that computes the function $f$ almost-uniquely if and only if there exists a PCM $N$ fulfilling $\#	ext{acc}_N(x) = f(x)$ for every $x \in \Sigma^*$.

**PROOF.** (1) is obvious. For (2), let $N$ be a PCM such that smallest and largest computation paths of $N$ are rejecting, and $N$ is time-bounded by polynomial $p$. Define $K$ to be the machine that, on input $x$, (a) guesses paths $y_1$ and $y_2$ with $y_1 < y_2$, $|y_1| = |y_2| = p(|x|)$, (b) simulates successively $N(x)$ on $y_1$ (result $\alpha_1$) and on $y_1 + 1$ (result $\beta_1$), (c) simulates successively $N(x)$ on $y_2$ (result $\alpha_2$) and on $y_2 + 1 \mod 2^p(|x|)$ (result $\beta_2$), and (d) outputs $y_2 - y_1$ and accepts iff $(\alpha_1 \oplus \beta_1) \land (\alpha_2 \oplus \beta_2)$ is true. Clearly, $K$ is an NPTM, and since $\#	ext{acc}_N(x) > 0$ we get $\#	ext{acc}_K(x) = 1$ and $\text{out}_K^+(x) = \{f(x)\}$. (3) combines (1) and (2).

**Corollary 3** For every set $A \in \text{NP}$ there is a PCM accepting $A$ in the sense of NP if and only if $\text{NP} = \text{UP}$.

The following results concern the class of counting functions computed by PCMs.

**Definition 4** $c\#P = \{\#	ext{acc}_M | M$ is a PCM$\}$.

The class UPSV [1] consists of all partial functions $f$ such that there is an NPTM $M$ with at most one accepting path, $\#	ext{acc}_M(x) = 1$ iff $f(x)$ is defined, and $\#	ext{acc}_M(x) = 1$ implies $\text{out}_M^+(x) = \{f(x)\}$. UPSV is the class of all total functions in UPSV.

**Theorem 5** (1) UPSV $\subseteq c\#P$.

(2) UPSV $= c\#P$ if and only if $\text{UP} = \text{coUP}$.

(3) A function $f$ is in UPSV $\iff$ its incremented version $f'$, defined as $f'(x) = f(x) + 1$, is in $c\#P$.

**PROOF.** (1) and (3) are immediate from Theorem 2. For (2), first, suppose UPSV $= c\#P$. Let $A \in \text{UP}$, then $c_A \in c\#P \subseteq$ UPSV (the latter via NPTM $M$). Modifying $M$ so that on unique accepting paths it is accepted iff output is 0, yields $A \in \text{coUP}$. Conversely, suppose UP $= \text{coUP}$. Let $f \in c\#P$. Define $A = \{x | f(x) \geq 1\}$. Then $A, \overline{A} \in \text{UP}$. Consider an NPTM $M$ that, on input $x$,
on paths beginning with 0 decides $x \in A$, on paths beginning with 1 decides $x \in \overline{A}$, if the accepting path gives $x \in A$ then outputs $f(x)$, and if the accepting path gives $x \in \overline{A}$ then outputs 0. Thus, $f \in \text{UPSV}_t$. □

Closure properties for complexity-theoretic function classes were first studied in [8]: A closure property is a function $f: \mathbb{N}^n \rightarrow \mathbb{N}$ for $n \in \mathbb{N}$, an $f \in \mathcal{F}$ is called an $\mathcal{F}$-closure property. A class $\mathcal{F}$ is closed under $f$ iff for all $g_1, \ldots, g_n \in \mathcal{F}$ the function $h$ in $\mathcal{F}$ where $h(x) = f(g_1(x), \ldots, g_n(x))$. $\mathcal{F}$ is called $\mathcal{G}$-closed iff $\mathcal{F}$ is closed under all $\mathcal{G}$-closure properties. A function $f \in \mathcal{F}$ is $\mathcal{F}$-hard whenever $\mathcal{F}$ is closed under $f$ iff $\mathcal{F}$ is $\mathcal{F}$-closed.

**Theorem 6** The following statements are equivalent.

1. $\text{UP} = \text{coUP}$.
2. $\text{c} \# \text{P}$ is closed under incrementation.
3. $\text{c} \# \text{P}$ is $\text{FP}$-closed.
4. $\text{c} \# \text{P}$ is $\text{c} \# \text{P}$-closed.

**PROOF.** Use Theorem 5 and note that $\text{UPSV}_t$ is $\text{UPSV}_t$-closed. □

It is well known that strong collapses of complexity classes such as $\text{PP} = \text{UP}$, $\text{PP} = \text{NP}$, or $\text{P}^{\text{PP}} = \text{NP}$, can be characterized by closure properties of some function classes (see [8,3]). The preceding theorem shows that, surprisingly, the very weak collapse $\text{UP} = \text{coUP}$ has such a characterization, too.

**Corollary 7** There is relativized world in which $\text{c} \# \text{P}$ is not $\text{c} \# \text{P}$-closed.

**PROOF.** Observe that relativizably, $\text{UP} = \text{coUP}$ implies $\text{NP}^{\text{UP}} = \text{NP}$. However in [10], an oracle was given which contradicts this equality. Note that all our proofs relativize. □

A function $f: \mathbb{N}^n \rightarrow \mathbb{N}$ is said to be 0-reproducing in its $k$-th component iff $m_k = 0$ implies $f(m_1, \ldots, m_n) = 0$ for all $m_1, \ldots, m_{k-1}, m_{k+1}, \ldots, m_n \in \mathbb{N}$. Moreover, $f$ is 0-reproducing iff $f$ is 0-reproducing in every component.

**Theorem 8** Let $f$ be an $n$-ary $\text{c} \# \text{P}$-closure property.

1. If $f$ is constant or 0-reproducing, then $\text{c} \# \text{P}$ is closed under $f$.
2. Let $h_1, \ldots, h_{k-1}, h_{k+1}, \ldots, h_n \in \text{FP}$. If $f$ is 0-reproducing in its $k$-th component, then $\text{c} \# \text{P}$ is closed under $\lambda x. f(h_1(x), \ldots, h_{k-1}(x), x, h_{k+1}(x), \ldots, h_n(x))$. 

5
**Proof.** (1): The case if \( f \) is constant is trivial. So, let \( f \) be \( 0 \)-reproducing, let \( g_1, \ldots, g_n \in \text{cP} \) be almost-uniquely computed by NPTMs’ \( M_1, \ldots, M_n \), and let \( D \) be a PCM with \( f = \text{acc} \). Define \( h(x) = f((g_1(x), \ldots, g_n(x)), \) and consider a machine \( N \) that, on \( x \), (a) simulates \( M_i(x) \) successively for all \( i \) in increasing order, (b) computes \( \langle z_1, \ldots, z_n \rangle \) if all simulations in (a) accept, where \( z_i \) is the output of \( M_i(x) \) on the accepting path, and (c) simulates \( D \) on \( \langle z_1, \ldots, z_n \rangle \). Since there is always at most one path where all simulations of \( M_i \) accept, \( N \) is a PCM with \( h(x) = \#\text{acc}_N(x) \). Thus \( h \in \text{cP} \). (2) is similar to (1). □

As an example, Theorem 8 immediately gives that \( \text{cP} \) is closed under division. This should be contrasted to the results for \( \text{P} \) in [8].

**Theorem 9** Let \( f \) be an \( n \)-ary \( \text{cP} \)-closure property. If there are \( k \in \mathbb{N} \) with \( 1 \leq k \leq n \), and \( x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n \in \mathbb{N} \) such that \( \lambda x. f(x_1, \ldots, x_{k-1}, x, x_{k+1}, \ldots, x_n) \) is neither \( 0 \)-reproducing nor constant, then \( f \) is \( \text{cP} \)-hard.

**Proof.** Suppose \( k, x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n \) make the premise true. To prove that \( f \) is \( \text{cP} \)-hard, assume \( \text{cP} \) is closed under \( f \). We have to show that \( \text{cP} \) is \( \text{cP} \)-closed. So, let \( \varphi(x) = f(x_1, \ldots, x_{k-1}, x, x_{k+1}, \ldots, x_n) \). Thus, \( \text{cP} \) is closed under \( \varphi \). Since \( \varphi \) is not \( 0 \)-reproducing, \( 0 < \varphi(0) = c_1 \). Since \( \varphi \) is not constant, there is a minimal \( z_0 \) so that \( c_1 \neq \varphi(z_0) = c_2 \). Defining \( \psi(x) = \varphi(\min(z_0, z_0 x)) \), we have \( x = 0 \Rightarrow \psi(x) = c_1 \) and \( x > 0 \Rightarrow \psi(x) = c_2 \). Since multiplication and minimization are \( 0 \)-reproducing and since \( \text{cP} \) is closed under \( \varphi \), \( \text{cP} \) is closed under \( \psi \). Now, there are two cases depending on \( c_2 \). If \( c_2 = 0 \), then define \( t = \text{sgn} \psi \), i.e., \( x = 0 \Rightarrow t(x) = 1 \) and \( x > 0 \Rightarrow t(x) = 0 \). Observe that \( \text{sgn} \) is \( 0 \)-reproducing. If \( c_2 > 0 \), then \( \psi(x) > 0 \) for all \( x \), thus \( \psi \) is almost-uniquely computable, and there is \( t \in \text{cP} \) with \( x = 0 \Rightarrow t(x) = 1 \) and \( x > 0 \Rightarrow t(x) = 0 \). In both cases, \( \text{cP} \) is closed under \( t \), and \( t \in \text{cP} \). Now, let \( A \in \text{UP} \). Then, \( c_A(x) = 1 \Leftrightarrow t(c_A(x)) = 0 \) and \( c_A(x) = 0 \Leftrightarrow t(c_A(x)) = 1 \). Hence, \( A \in \text{coUP} \). Thus, \( \text{UP} = \text{coUP} \) and using Theorem 6 we get that \( \text{cP} \) is \( \text{cP} \)-closed. □

Though we do not know whether \( \text{cP} \) is \( \text{cP} \)-closed, Theorem 8 and Theorem 9 yield a criterion to decide whether a unary function \( f \in \text{cP} \) is hard for \( \text{cP} \) or not: If \( f \) is neither \( 0 \)-reproducing nor constant, then it is hard, otherwise \( \text{cP} \) is closed under \( f \).

**Corollary 10** For every unary \( \text{cP} \)-closure property \( f \), it holds that \( \text{cP} \) is closed under \( f \) in all relativizations if and only if \( f \) is \( 0 \)-reproducing or constant.
4 A Taxonomy of Clustering in Counting and Optimization

In this section we generalize our definitions to arbitrary sets. For that, let $A \subseteq \Sigma^*$, and let $A_x = \{ y | \langle x, y \rangle \in A \}$. For $x, y, z \in \Sigma^*$, we write $y \equiv_x z$ iff $|y - z| = 1$ and $\langle x, y \rangle \in A \Leftrightarrow \langle x, z \rangle \in A$. The relation $\equiv_x^*$ is defined as the equivalence relation generated by $\equiv_x$. A is said to be a cluster set iff for every
there is an $y$ such that $A_x = \{ z | z \equiv y \land \langle x, y \rangle \in A \}$.

The unambiguous cluster versions of the ordinary operators are obtained if we allow the operators to vary only over cluster sets in a class $\mathcal{K}$. A function $f$ is in $c \# \cdot \mathcal{K}$ iff there are a cluster set $A \in \mathcal{K}$ and a polynomial $p$ so that $f(x) = \#A_x^p$ for every $x$. Evidently, $c \# \cdot \mathcal{P} = c \# \mathcal{P}$. The definitions of $\text{cmax}$ and $\text{cmin}$ are similar.

Note that, though the value $f(x)$ above depends on $A_x^p$, we require that $A$ is a cluster set, not only the set $A^p$ defined as $\{ \langle x, y \rangle | \| y \| \leq p(|x|) \land \langle x, y \rangle \in A \}$. Generally, to require that only $A^p$ is a cluster set leads to larger classes.

However, for classes $\mathcal{K}$ closed under $\leq_m^p$-reductions or closed under intersection with $\mathcal{P}$-sets, both approaches are equivalent.

**Theorem 11** All inclusions, equivalences, and implications presented in Fig. 3 hold.

**Proof.** The techniques are straightforward. Thus we only prove one result for each type of statement as examples. The remaining claims are left to the reader. Observe from [4] that for complexity classes $\mathcal{C}, \mathcal{K}$ closed under $\leq_m^p$-reductions, it holds $\text{cmin} \cdot \mathcal{C} \subseteq \text{cmax} \cdot \mathcal{K} \iff \text{cmax} \cdot \mathcal{C} \subseteq \text{cmin} \cdot \mathcal{K}$.

(I) Inclusions. $c \# \cdot \text{NP} \subseteq \text{cmax} \cdot \text{NP}$: Let $f \in c \# \cdot \text{NP}$ by $A \in \text{NP}$ and polynomial $p$. Consider $B = \{ \langle x, y \rangle | \| y \| = p(|x|) \land (\exists z_1, z_2, |z_1| = |z_2| = p(|x|)) \} \cap \{ \langle x, z \rangle | \langle x, z \rangle \in A \land x \neq y = |z_1 - z_2| + 1 \}$. Then $B \in \text{NP}$ and $B = \{ \langle x, y \rangle | \| y \| = p(|x|) \land y \leq f(x) \}$. Thus, $B$ is a cluster set and $f(x) = \max B_x^p$. Hence, $f \in \text{cmax} \cdot \text{NP}$.

(II) Equivalences. $\text{PH} = \text{UP} \iff \text{cmax} \cdot \text{NP} = c \# \cdot \mathcal{P}$: For $[\Rightarrow]$ suppose $\text{PH} = \text{UP}$. Then conclude: $c \# \cdot \mathcal{P} \subseteq \text{cmax} \cdot \text{NP} \subseteq \text{cmax} \cdot \mathcal{P} = \text{cmin} \cdot \mathcal{P} = \text{cmax} \cdot \text{NP}$. An easy analysis shows that the latter class is equal to $c \# \cdot \mathcal{P}$. For $[\Leftarrow]$ suppose $\text{cmax} \cdot \text{NP} = c \# \cdot \mathcal{P}$. We show $\text{coNP} \subseteq \text{UP}$. Let $A \in \text{NP}$ accepted by NPTM $M$. Define $N$ to be the machine that, on input $x$, simulating $M(x)$ outputs 1 on rejecting paths, outputs 2 on accepting paths, and always accepts. Note that $\text{out}^N_K(x) \subseteq \{1, 2\}$ and $\langle x, y \rangle \in \text{out}^N_K(x) \in \text{NP}$. Set $f(x) = \max \text{out}^N_K(x)$. Then, $f \in \text{cmax} \cdot \text{NP}$ and $x \in A \Rightarrow f(x) = 1$ and $x \notin A \Rightarrow f(x) = 2$. Since $\text{cmax} \cdot \text{NP} = c \# \cdot \mathcal{P}$, there is an NPTM $K$ that computes $f$ uniquely. Thus $K$ can be modified to accept $A$ in the sense of UP.

(III) Implications. Use the monotonic operator technique (see [12,4]). For a function class $\mathcal{F}$, say $A \in \mathcal{U} \cdot \mathcal{F}$ iff there is an $f \in \mathcal{F}$ so that for every $x$, $x \in A \iff f(x) = 1$ and $x \notin A \iff f(x) = 0$. Say $A \in \exists \cdot \mathcal{F}$ iff an $f \in \mathcal{F}$ exists satisfying $x \in A \iff f(x) \geq 1$. With $\# \cdot \mathcal{P} = \# \cdot \text{coNP} [6]$, it is easily seen that $\text{UP} = \mathcal{U} \cdot \# \cdot \mathcal{P} = \mathcal{U} \cdot \# \cdot \text{coNP} \subseteq (\exists \cdot \text{cmax} \cdot \text{coNP} \cap \exists \cdot \text{cmin} \cdot \text{coNP} \cap \exists \cdot c \# \cdot \text{coNP})$. Further, $\exists \cdot \text{FP} = \exists \cdot \text{cmax} \cdot \text{NP} = \text{NP}$.
Main open problems concern the position of $c\# \cdot \text{coNP}$ in the taxonomy of the figure. Especially, it is not known whether $c\# \cdot \text{P}^{\text{NP}} = c\# \cdot \text{coNP}$. As already mentioned this is true for the usual counting operator [6]. A partial answer gives the next theorem.

**Theorem 12** The following statements are equivalent.

1. $c\# \cdot \text{P}^{\text{NP}} = c\# \cdot \text{coNP}$.
2. $\text{cmax} \cdot \text{P}^{\text{NP}} \subseteq c\# \cdot \text{coNP}$.
3. $\text{FP}^{\text{NP}} \subseteq c\# \cdot \text{coNP}$.

**PROOF.** [(1) \Rightarrow (2)] and [(2) \Rightarrow (3)] are trivial. For [(3) \Rightarrow (1)] suppose $\text{FP}^{\text{NP}} \subseteq c\# \cdot \text{coNP}$. Let $f \in c\# \cdot \text{P}^{\text{NP}}$. From relativized Theorem 2 we obtain an NPTM $M$ having oracle access and $A \in \text{NP}$, such that $\#\text{acc}_{M^{(A)}}(x) \leq 1$ and $\#\text{acc}_{M^{(A)}}(x) = 1 \Rightarrow f(x) = \text{out}_{M^{(A)}}^{+}(x)$. Define $g(x,y) = \text{out}_{M^{(A)}}^{+}(x)$ if $y \in \text{acc}_{M^{(A)}}(x)$ and $g(x,y) = 0$ otherwise. Then $g \in \text{FP}^{\text{NP}}$. Consequently, there are a cluster set $C \in \text{coNP}$ and polynomials $p$ and $r$ with $g(x,y) = \#\{z||z| = r(|x|) \land |y| = p(|x|) \land \langle x, y, z \rangle \in C\}$. Note that for at most one $y$ it holds $g(x,y) > 0$, when $x$ is given. Define $C' = \{(x,y) ||y| = r(|x|) + p(|x|) \land \langle x, y, z \rangle \in C\}$. Then, $C' \in \text{coNP}$, and $C'$ is a cluster set. Define $g'(x) = \#\{w||w| = r(|x|) + p(|x|) \land \langle x, w \rangle \in C'\}$. Then, $g' \in c\# \cdot \text{coNP}$ and $g'(x) = g(x,y) = f(x)$ if such $y$ exists, otherwise $g'(x) = f(x) = 0$. Hence, $f \in c\# \cdot \text{coNP}$. □

Next we compare cluster classes and their corresponding ambiguous classes. Trivially, the unambiguous classes are subclasses of the ambiguous one’s.

**Theorem 13** (1) $c\# \cdot \text{P} = \# \cdot \text{P} \iff \text{PP} = \text{UP}$.
(2) $c\# \cdot \text{NP} = \# \cdot \text{NP} \iff \text{PP} = \text{NP}$.
(3) If $c\# \cdot \text{coNP} = \# \cdot \text{coNP}$ then $\text{PP} \subseteq \text{UP}^{\text{NP}}$.

**PROOF.** (1): For $[\Rightarrow]$ suppose $c\# \cdot \text{P} = \# \cdot \text{P}$. Let $A \in \text{PP}$ via $f \in \# \cdot \text{P}$ and $g \in \text{FP}$ with $x \in A \iff f(x) \geq g(x)$ and $f(x) > 0$ for every $x$. Then, $f \in c\# \cdot \text{P}$, i.e., there is an NPTM $M$ that computes $f$ uniquely. Modify $M$ such that $M$ accepts on the only accepting path if $f(x) \geq g(x)$. Then $A \in \text{UP}$.

For $[\Leftarrow]$ suppose $\text{PP} = \text{UP}$, equivalently C\_P = UP [8]. Let $f \in \# \cdot \text{P}$. Define $C = \{(x,y)||y| = p(|x|) \land f(x) = y\}$. Then $C \in \text{C\_P} \subseteq \text{UP}$. Define $N$ to be a machine that, on input $x$, checks $\langle x, y \rangle \in C$ for a guessed word $y$, $|y| = p(|x|)$, outputs $y$, and accepts iff checking is successful. $N$ computes $f$ uniquely. Thus, $f \in c\# \cdot \text{P}$.

(2): For $[\Rightarrow]$ suppose $c\# \cdot \text{NP} = \# \cdot \text{NP}$. Let $A \in \text{PP}$ via $f \in \# \cdot \text{P} \subseteq \# \cdot \text{NP}$ and $g \in \text{FP}$ with $x \in A \iff f(x) \geq g(x)$ and $f(x) > 0$ for every $x$. Then,
\[ f \in c\# \cdot NP = \text{cmax} \cdot \text{NP}, \text{i.e., there exist a cluster set} \ B \in \text{NP}, \text{accepted by}\ \text{NPTM} \ M, \text{and polynomial} \ p \text{ with} \ f(x) = \max \{ y||y| = p(|x|) \wedge \langle x, y \rangle \in B \}. \text{Define} \ N \text{ to be a machine that, on input} \ x, \text{simulates} \ M \text{ on input} \ \langle x, y \rangle \text{ with} \ y \text{ being a guessed word,} \ |y| = p(|x|), \text{computes deterministically} \ g(x), \text{and accepts iff} \ M(\langle x, y \rangle) \text{ is accepting on the simulation path and it holds} \ y \geq g(x). \text{Thus,} \ A \in \text{NP}. \text{For} \ \llbracket \supseteq \rrbracket \text{ suppose} \ \text{PP} = \text{NP} \text{ and conclude} \ c\# \cdot \text{NP} \subseteq \# \cdot \text{NP} \subseteq c\# \cdot \text{P}\#^{NP} \subseteq c\# \cdot \text{CH} = c\# \cdot \text{NP} \text{ since} \ \text{PP} = \text{NP} \iff \text{CH} = \text{NP} \text{ (corollary of Toda’s Theorem [11]).} \]

(3) is due to \# \cdot \text{P}^{NP} = \# \cdot \text{coNP} \text{ and to relativization of (1).} \]
References


