*Electronic Colloquium on Computational Complexity* - Reports Series 1996 - available via:
ECCC
TR96-045
FTP: ftp.eccc.uni-trier.de:/pub/eccc/
WWW: http://www.eccc.uni-trier.de/eccc/
Email: ftpmail@ftp.eccc.uni-trier.de with subject 'help eccc'

# Powers-of-Two Acceptance Suffices for Equivalence and Bounded Ambiguity Problems

*Bernd Borchert*[1]
Mathematisches Institut
Universität Heidelberg
69120 Heidelberg, Germany

*Lane A. Hemaspaandra*[2]
Department of Computer Science
University of Rochester
Rochester, NY 14627, USA

*Jörg Rothe*[3]
Institut für Informatik
Friedrich-Schiller-Universität Jena
07743 Jena, Germany

August 28, 1996

[1]Email: bb@math.uni-heidelberg.de.

[2]Research supported in part by grants NSF-CCR-9322513 and NSF-INT-9513368/DAAD-315-PRO-fo-ab. Work done in part while visiting Friedrich-Schiller-Universität Jena. Email: lane@cs.rochester.edu.

[3]Research supported in part by grant NSF-INT-9513368/DAAD-315-PRO-fo-ab. Work done in part while visiting the University of Rochester and Le Moyne College. Email: rothe@mipool.uni-jena.de.

**Abstract**

We study EP, the subclass of NP consisting of those languages accepted by NP machines that when they accept always have a number of accepting paths that is a power of two. We show that the negation equivalence problem for OBDDs (ordered binary decision diagrams [FHS78,Bry92]) and the interchange equivalence problem for 2-dags are in EP. We also show that for boolean negation [Har71] the equivalence problem is in $EP^{NP}$, thus tightening the existing $NP^{NP}$ upper bound. We show that FewP [All86,AR88], bounded ambiguity polynomial time, is contained in EP, a result that seems incomparable with the previous SPP upper bound. Finally, we show that EP can be viewed as the promise-class analog of $C_{=}P$.

# 1 Introduction

NP languages can be defined via machines that reject by having zero accepting paths, and that accept by having their number of accepting paths belong to the set $\{1, 2, 3, \cdots\}$. A number of researchers have sought to refine the class NP by shrinking the path-cardinality set signifying acceptance, while retaining the requirement that rejection be associated with having zero accepting paths. Valiant's class UP [Val76], which is known to differ from P if and only if one-way functions exist [GS88,Ko85], has the acceptance set $\{1\}$. Acceptance sets of the forms $\{1, 2, 3, \cdots, n^{\mathcal{O}(1)}\}$ and $\{1, 2\}$, $\{1, 2, 3\}$, $\cdots$ define, respectively, the class FewP [All86,AR88] and the classes $\mathrm{UP}_{\leq 2}$, $\mathrm{UP}_{\leq 3}$, $\cdots$ [Wat88,Bei89] (note: $\mathrm{UP} \subseteq \mathrm{UP}_{\leq 2} \subseteq \mathrm{UP}_{\leq 3} \subseteq \cdots \subseteq \mathrm{FewP} \subseteq \mathrm{NP}$). These classes are also connected to the existence of one-way functions and have been studied in a wide variety of contexts, such as in terms of class containments, complete sets, reducibilities, boolean hierarchy equivalences, and upward separations (see, e.g., [KSTT92,HJV93,FFK94,HH94,HR,RRW94,Rot95]). Of course, the litmus test of NP refinements such as UP, $\mathrm{UP}_{\leq k}$, and FewP is *the extent to which they allow us to refine the upper bounds on the complexity of natural* NP *problems.* Of these classes, UP has been most successful in this regard. UP is known to provide an upper bound on the complexity of (a language version of) the discrete logarithm problem [GS88], and UP (indeed $\mathrm{UP} \cap \mathrm{coUP}$) is known to provide an upper bound on the complexity of primality testing [FK92].

However, there are certain NP problems whose richness of structure has to date defied attempts to put them in UP or even FewP, yet that nonetheless intuitively seem to use less than the full generality of NP's acceptance mechanism. To try to categorize these problems, we introduce the class EP, which is intermediate between FewP and NP: $\mathrm{FewP} \subseteq \mathrm{EP} \subseteq \mathrm{NP}$. In particular, EP is the NP subclass whose acceptance set is $\{2^i \mid i \in \mathbb{N}\}$, $\mathbb{N} = \{0, 1, 2, 3, \cdots\}$.

In Section 2, we provide improved upper bounds on the complexity of OBDD (Ordered Binary Decision Diagram) Negation Equivalence, 2-Dag Interchange Equivalence, and Boolean Negation Equivalence. These three problems are trivially in, respectively, NP, NP, and $\mathrm{NP}^{\mathrm{NP}}$. We provide, respectively, EP, EP, and $\mathrm{EP}^{\mathrm{NP}}$ upper bounds. The problems are not known to belong to (and do not seem to obviously belong to), respectively, FewP, FewP, and $\mathrm{FewP}^{\mathrm{NP}}$. In Section 3, we prove that EP contains FewP, and we discuss the location of the class EP with respect to other complexity classes. In Section 4, we discuss the "non-promise" analog of EP: the class having EP's acceptance set ($\{2^i \mid i \in \mathbb{N}\}$) but having the rejection set $\mathbb{N} - \{2^i \mid i \in \mathbb{N}\}$. We show that this change boosts EP's power to

exactly that of C$_=$P, where C$_=$P [Sim75,Wag86] is the class of all sets $L$ such that there is a polynomial-time function $f$ and a nondeterministic polynomial-time Turing machine $N$ such that for each $x$, $x \in L$ if and only if $N(x)$ has exactly $f(x)$ accepting paths. This provides a motivation for EP—namely that EP is a promise analog of C$_=$P—that differs from the motivation implicitly provided by the upper-bound examples of Section 2.

## 2 Concrete Problems and EP

In this section, we provide concrete problems known to be in NP (or NP$^{\mathrm{NP}}$), and we prove they are in fact in EP (or EP$^{\mathrm{NP}}$). We now introduce the class EP (mnemonic: the number of paths is restricted to being either 0 or some power (some exponention) of 2).

**Definition 2.1** EP *denotes the class of all languages $L$ for which there is a nondeterministic polynomial-time Turing machine $N$ such that, for each input $x \in \Sigma^*$,*

$$\begin{aligned} x \notin L \quad &\Longrightarrow \quad \#acc_N(x) = 0, \text{ and} \\ x \in L \quad &\Longrightarrow \quad \#acc_N(x) \in \{2^i \mid i \in \mathbb{N}\}. \end{aligned}$$

*The natural operator analog of* EP *is defined as follows. For any class $\mathcal{K}$, let $\mathbf{E} \cdot \mathcal{K}$ denote the class of all languages $L$ for which there exist a set $A$ in $\mathcal{K}$ and a polynomial $p$ such that, for each $x \in \Sigma^*$,*

$$\begin{aligned} x \notin L \quad &\Longrightarrow \quad ||\{y \mid |y| = p(|x|) \ \wedge \ \langle x, y\rangle \in A\}|| = 0, \text{ and} \\ x \in L \quad &\Longrightarrow \quad ||\{y \mid |y| = p(|x|) \ \wedge \ \langle x, y\rangle \in A\}|| \in \{2^i \mid i \in \mathbb{N}\}. \end{aligned}$$

Clearly, $\mathbf{E} \cdot \mathrm{P} = \mathrm{EP}$. Consider the following problem.

`Problem:` Boolean Negation Equivalence (BNE) (see the survey by Harrison [Har71] and the bibliography provided after the references in the paper by Borchert, Ranjan, and Stephan [BRS95])
`Input:` Two boolean functions (input as Boolean formulas using variable names and the symbols $\{\wedge, \vee, \neg, (, )\}$), $f(x_1, \cdots, x_n)$ and $g(x_1, \cdots, x_n)$, over the same $n$ boolean variables.[1]
`Question:` Are $f$ and $g$ negation equivalent? That is, can one negate some of the inputs of $g$ such that $f$ and the modified function $g'$ are the same. For concreteness as a language problem, BNE $= \{(f, g) \mid f$ and $g$ are negation equivalent$\}$.

[1] The operator set $\{\wedge, \vee, \neg\}$ is specified for concreteness. Any other complete set of operations would do, e.g., {nxor}.

For example, the two boolean functions described by the formulas $x_1 \vee x_2 \vee x_3$ and $x_1 \vee \neg x_2 \vee \neg x_3$ are negation equivalent by negating $x_2$ and $x_3$. Regarding lower bounds, Borchert, Ranjan, and Stephan [BRS95] have shown that BNE is US-hard [BG82], and thus in particular is coNP-hard. Regarding upper bounds, BNE $\in \mathrm{NP}^{\mathrm{NP}}$ [BRS95] and BNE $\in \mathrm{coAM}^{\mathrm{NP}}$ (combining [BRS95] and [AT96]). It follows from the latter that BNE is not $\mathrm{NP}^{\mathrm{NP}}$-complete unless the polynomial hierarchy collapses ([AT96], in light of [BRS95, Sch89]). Interestingly, these two upper bounds—$\mathrm{NP}^{\mathrm{NP}}$ and $\mathrm{coAM}^{\mathrm{NP}}$—seem incomparable. We now prove BNE $\in \mathrm{EP}^{\mathrm{NP}}$, which is probably incomparable with the $\mathrm{coAM}^{\mathrm{NP}}$ upper bound but which improves the $\mathrm{NP}^{\mathrm{NP}}$ upper bound, as clearly $\mathrm{EP}^{\mathrm{NP}} \subseteq \mathrm{NP}^{\mathrm{NP}}$.

**Theorem 2.2** BNE $\in \mathrm{EP}^{\mathrm{NP}}$ (*indeed,* BNE $\in \mathbf{E} \cdot \mathrm{coNP}$).

**Proof.** It is obvious that negation equivalence is an equivalence relation on boolean functions. However, negation equivalence has much stronger properties that will be helpful in this proof. First, we remind the reader of some basic definitions from linear algebra (see, e.g., [Smi84]). Let $\mathcal{V}$ be a vector space over some field $\mathcal{F}$. A nonempty subset $\mathcal{U}$ of $\mathcal{V}$ is called a *linear subspace* of $\mathcal{V}$ if both (i) if $\vec{a} \in \mathcal{U}$ and $\vec{b} \in \mathcal{U}$ then $\vec{a} + \vec{b} \in \mathcal{U}$, and (ii) if $\vec{a} \in \mathcal{U}$ and $c \in \mathcal{F}$ then $c\vec{a} \in \mathcal{U}$. If $\mathcal{V}$ is a vector space and $\mathcal{U}$ is a linear subspace of $\mathcal{V}$ and $\vec{a} \in \mathcal{V}$, then the set $\{\vec{a} + \vec{b} \mid \vec{b} \in \mathcal{U}\}$ is called an *affine subspace of* $\mathcal{V}$.

Suppose a given instance of BNE consists of $f$ and $g$, each over the variables $x_1, \cdots, x_n$. A negation of some of the input variables of $g$ as in the definition of BNE can be represented by a vector $\vec{v} = (c_1, \ldots, c_n)$ in the vector space $\mathrm{GF}(2)^n$, where each $c_i$ is either 0 or 1 and $c_i = 1$ means that the variable $x_i$ will be negated. Let $g_{\vec{v}}$ be the boolean function resulting from $g$ after the application of the negations described by $\vec{v}$, i.e., $g_{\vec{v}}(\vec{u}) = g(\vec{v} + \vec{u})$. Now it is easy to see (double negation equals identity) that, for each fixed boolean function $g$, the set of negation vectors $\vec{v}$ such that $g$ equals $g_{\vec{v}}$ is a linear subspace $V_g$ of $\mathrm{GF}(2)^n$. It is not hard to see that if $\vec{w}$ is *any* negation vector such that $f = g_{\vec{w}}$, then the affine subspace $\vec{w} + V_g$ is the set of *all* negation vectors witnessing the negation equivalence of $f$ and $g$. Why? It is clear that each element of $\vec{w} + V_g$ is *a* negation vector witnessing the negation equivalence of $f$ and $g$. Suppose there is some negation vector, $\vec{z}$, such that $\vec{z} \notin \vec{w} + V_g$ and yet $\vec{z}$ is a negation vector witnessing the negation equivalence of $f$ and $g$. Since $\vec{w}$ and $\vec{z}$ are both negation vectors witnessing the negation equivalence of $f$ and $g$, we have $f = g_{\vec{w}}$ and $f = g_{\vec{z}}$. So $g_{\vec{w}} = g_{\vec{z}}$. But we are working in $\mathrm{GF}(2)^n$, so this says that, for any $\vec{u}$, $g(\vec{w} + \vec{u}) = g(\vec{z} + \vec{u})$, so setting $\vec{u} = \vec{w} + \vec{u'}$ and substituting, we have $g(\vec{u'}) = g_{\vec{w}+\vec{z}}(\vec{u'})$. So $\vec{w} + \vec{z} \in V_g$, and thus $\vec{z} \in \vec{w} + V_g$, contradicting our assumption that $\vec{z} \notin \vec{w} + V_g$. (This type

of argument is familiar from the graph analog; once one knows the automorphism group $G$ of a graph and one isomorphism $i$ from some other graph to that graph, then the set of all isomorphisms is well-known to be given by $G \circ i$.)

Of course, $\vec{w} + V_g$ will be of the same cardinality as the subspace $V_g$ (as addition by $\vec{w}$ induces a bijection between $\mathrm{GF}(2)^n$ and itself), and as an $\ell$-dimensional vector space over the field $\mathrm{GF}(2)$ has exactly $2^\ell$ vectors, $\vec{w} + V_g$ will contain exactly $2^m$ vectors, where $m$ is the dimension of $V_g$.

So the following nondeterministic program shows that BNE is in EP with an NP oracle: Read the two input functions $f$ and $g$ (checking that they are both over the same number of variables and that the variables have the same naming scheme), guess a negation vector $\vec{v}$ and accept if and only if the oracle confirms that $f$ is equal to $g$ altered by the negation vector $\vec{v}$. This shows that BNE is in $\mathrm{EP}^{\mathrm{NP}}$, since if $f$ and $g$ are not negation equivalent, then there is no accepting path, and otherwise there are exactly $2^m$ accepting paths, where $m$ is the dimension of the affine subspace discussed above.

In fact, this shows BNE $\in \mathbf{E} \cdot \mathrm{coNP}$. ∎

We remark that the parenthetical part of Theorem 2.2 is not superfluous in any obvious way. That is, even though in the NP analog it would be superfluous as $\mathrm{NP}^{\mathrm{NP}} = \exists \cdot \mathrm{coNP}$, it is not at all clear that $\mathrm{EP}^{\mathrm{NP}} = \mathbf{E} \cdot \mathrm{coNP}$. However, clearly $\mathbf{E} \cdot \mathrm{coNP} \subseteq \mathrm{EP}^{\mathrm{NP}[1]} \subseteq \mathrm{EP}^{\mathrm{NP}}$, so the parenthetical remark is stronger, perhaps strictly, as it is not at all clear that NP is in $\mathbf{E} \cdot \mathrm{coNP}$.

There are ways of describing boolean functions such that the equivalence problem is in P. The most prominent such way is by ordered binary decision diagrams (OBDDs). (Fortune, Hopcroft, and Schmidt [FHS78] were the ones who proved that equivalence for OBDDs is in P. For complete background on OBDDs see, for example, the survey by Bryant [Bry92].) Thus, essentially by the discussion in the previous paragraph, the following computational problem, OBDD Negation Equivalence, is in (non-relativized) EP: Given a pair $(e, f)$ of OBDDs, are the boolean functions described by $e$ and $f$ negation equivalent?

If we consider the special case that for the two OBDDs $(e, f)$ above the order of the variables is required to be the same, we see that the following graph-theoretic problem is in (non-relativized) EP. Let a *2-dag* be a directed acyclic graph (without labels) with a unique root and either 0 or 2 ordered successors for each node. For a 2-dag each node is assigned a depth, namely the distance to the root. Now consider the following computational problem (2-Dag Interchange Equivalence): Given two 2-dags $F$ and $G$, is there a set of natural numbers $(i_1, \ldots, i_n)$ such that, if in $G$ for each node of depth $i_1, \ldots, i_n$ its two successors

(if they exist) are interchanged, then the modified 2-dag $G'$ equals $F$? This problem can be shown to be in EP (similarly to the argument above). Moreover, the problem can easily be reduced to the Graph Isomorphism problem. The authors know of no P algorithm for the general case of 2-Dag Interchange Equivalence, though the special case of this problem with binary trees instead of general 2-dags has an easy deterministic polynomial-time algorithm.

## 3 Location of EP

It is immediate from its definition that EP $\subseteq$ NP. It is also clear that the quantum-computation-related class (a definition is included later) $\mathrm{C_{=}P}$[half] of Berthiaume and Brassard [BB92] is also contained in EP. We now prove that FewP $\subseteq$ EP.

**Theorem 3.1** FewP $\subseteq$ EP.

**Proof.** Our proof technique builds on that used by Cai and Hemaspaandra (then Hemachandra) [CH90] to prove FewP $\subseteq \oplus$P, where $\oplus$P [GP86,PZ83] is the class of languages $L$ such that for some nondeterministic polynomial-time Turing machine machine $N$, on each $x$ it holds that $x \in L \iff \#acc_N(x) \equiv 1 \pmod 2$. Köbler, Schöning, Toda, and Torán [KSTT92] also built on that technique in their proof that FewP $\subseteq \mathrm{C_{=}P}$.

By definition, UP is in EP, and $\mathrm{UP}_{\leq 2}$ is in EP.

$\mathrm{UP}_{\leq 3}$ is in EP as follows. Given a $\mathrm{UP}_{\leq 3}$ machine $M$, define an EP machine $N$ as follows. We will choose positive constants $c_1$, $c_2$, and $c_3$ such that, for each $x$: (a) for each accepting path of $M(x)$, our new machine $N$ will have $c_1$ accepting paths on input $x$; (b) for each pair of accepting paths of $M(x)$, $N(x)$ will have $c_2$ additional accepting paths; and (c) for each triple of accepting paths of $M(x)$, $N(x)$ will have $c_3$ additional accepting paths. So for instance, if $M(x)$ has three accepting paths, then

$$\#\mathrm{acc}_N(x) = \binom{3}{1}c_1 + \binom{3}{2}c_2 + \binom{3}{3}c_3.$$

So now let us choose $c_1$, $c_2$, and $c_3$. Set $c_1$ to 1, which is a power of two. Set $c_2$ to be as small as possible consistent with ensuring that $\binom{2}{1}c_1 + c_2$ is a power of two, i.e., set $c_2 = 2$. Now set $c_3$ to be as small as possible consistent with ensuring $\binom{3}{1}c_1 + \binom{3}{2}c_2 + c_3$ is a power of two, i.e., set $c_3 = 7$. It follows that $\mathrm{UP}_{\leq 3}$ is in EP.

To prove that $\mathrm{UP}_{\leq 4}$ is in EP, we have to ensure that the number of accepting paths of our EP machine $N$ satisfies the new equation

$$\#\mathrm{acc}_N(x) = \binom{4}{1}c_1 + \binom{4}{2}c_2 + \binom{4}{3}c_3 + \binom{4}{4}c_4$$

for suitable constants $c_1, \ldots, c_4$. Choose $c_1$, $c_2$, and $c_3$ as before. Set $c_4$ to be as small as possible consistent with ensuring $\binom{4}{1}c_1 + \binom{4}{2}c_2 + \binom{4}{3}c_3 + c_4$ is a power of two. This is achieved by choosing $c_4 = 20$. It follows that $\mathrm{UP}_{\leq 4}$ is in EP.

It is clear that we can continue in this way, and this shows that $\mathrm{UP}_{\leq \mathcal{O}(1)}$ is in EP.

However, note that we can go beyond constants. In fact, we can go as far as we can get while ensuring that $c_{f(n)}$ is at most exponential. To prove that FewP is in EP, we must estimate $c_j$. We claim that for each $j$,

$$c_j \leq 2 \cdot j \cdot \binom{j}{\lceil \frac{j}{2} \rceil} \cdot c_{j-1}. \tag{3.1}$$

Why? The factor 2 gets us to the next power of two, since from any integer there is always a power of two that is no more than double that integer. The factor $j$ is the number of terms in the equation

$$\#\mathrm{acc}_N(x) = \binom{j}{1}c_1 + \binom{j}{2}c_2 + \binom{j}{3}c_3 + \cdots + \binom{j}{j-1}c_{j-1} + \binom{j}{j}c_j. \tag{3.2}$$

The coefficient $\binom{j}{\lceil \frac{j}{2} \rceil}$ is the biggest binomial coefficient of any term in Equation 3.2. Furthermore, $c_{j-1}$ is the largest constant among the $c_i$, $1 \leq i \leq j-1$, when we are seeking to determine $c_j$. Thus, Equation 3.1 gives a valid bound for $c_j$. Hence, if we can show that $c_{\mathrm{poly}}$ is in $\mathcal{O}(2^{\mathrm{poly}})$, we have shown that FewP is in EP.

From Equation 3.1 above and the fact that for every large enough $j$,

$$2 \cdot j \cdot \binom{j}{\lceil \frac{j}{2} \rceil} \leq \binom{j}{\lceil \frac{j}{2} \rceil}^2 \leq \left(4^j\right)^2,$$

and the fact that $c_1 = 1$, we can unwind the recurrence relation that is implicit in Equation 3.1, and we obtain

$$c_j \leq \prod_{1 \leq i \leq j} 4^{2i}.$$

So, adding the exponents,

$$c_j \leq 4^{\sum_{1 \leq i \leq j} 2i}.$$

Thus, $c_j \leq 4^{2\binom{j+1}{2}}$. This implies $c_j \leq 4^{\mathcal{O}(j^2)}$. Hence, $c_{\mathrm{poly}}$ is at most exponential. Of course, all the $c_1$, $c_2$, ..., $c_{\mathrm{poly}}$ are easy to compute in polynomial time. This completes the proof that FewP is contained in EP. ∎

An immediate question is how Theorem 3.1 relates to known results about FewP. Clearly, Theorem 3.1 represents an improvement on the trivial inclusion FewP $\subseteq$ NP.

However, how does it compare with the nontrivial result of Köbler et al. [KSTT92] and Fenner, Fortnow, and Kurtz [FFK94] that FewP $\subseteq$ Few $\subseteq$ SPP $\subseteq \oplus\mathrm{P} \cap \mathrm{C}_{=}\mathrm{P}$?[2] There are a number of related aspects to this question. First, is SPP $\subseteq$ EP, which would make Theorem 3.1 a trivial consequence of the known result FewP $\subseteq$ SPP? This seems unlikely, as if SPP $\subseteq$ EP, then SPP $\subseteq$ NP, and SPP $\subseteq$ NP is considered unlikely (see [FFK94,TO92]). Second, is EP $\subseteq$ SPP, which would make Theorem 3.1 a strengthening of the known result that FewP $\subseteq$ SPP? We do not know. Third, notice that we proved FewP $\subseteq$ EP but that the Köbler et al. [KSTT92] and Fenner, Fortnow, and Kurtz [FFK94] work shows that Few $\subseteq$ SPP. Can our result be extended to show Few $\subseteq$ EP? The reason we mention this is that often it is the case that when one can prove something about FewP, then one can also prove it about the slightly bigger class Few. For example, Cai and Hemaspaandra, after showing that FewP is in $\oplus$P, then easily applied their technique to show that even Few is in $\oplus$P [CH90]. Similarly, it is immediately clear that FewP has Turing-complete sets if and only if Few has Turing-complete sets, and so the proof that there is a relativized world in which FewP lacks Turing-complete sets [HJV93] implicitly proves that there is a world in which Few lacks Turing-complete sets (see also [Ver94]). However, in the case of Theorem 3.1, it is unlikely that by modifying the technique in a way similar to that done by Cai and Hemaspaandra one could hope to establish the slightly stronger result that EP even contains Few. Why? Clearly coUP $\subseteq$ Few and EP $\subseteq$ NP, so the assumption Few $\subseteq$ EP would imply (along with other even more unlikely things) coUP $\subseteq$ NP.

Fourth, one might wonder directly, since FewP $\subseteq \oplus$P is known, about the relationship between EP and $\oplus$P. That is, how is EP (powers-of-two acceptance) related to $\oplus$P (multiples-of-two acceptance).[3] We note the following. By a diagonalization so routine as to not be worth including here, one can show $(\exists A)\,[\mathrm{coUP}^A \not\subseteq \mathrm{EP}^A]$. It follows immediately that $(\exists A)\,[\mathrm{FewP}^A \not\subseteq \mathrm{EP}^A]$ and $(\exists A)\,[\oplus\mathrm{P}^A \not\subseteq \mathrm{EP}^A]$. Similarly, if one looks at the test language inside the proof of Proposition 12 of Beigel's 1991 "mod classes" paper [Bei91], one can see that for his case "$k = 2$" the test language is in (relativized) $\mathrm{C}_{=}\mathrm{P}$[half],[4]

[2] Informally stated, Few [CH90] is what a P machine can do given one call to a #P function that obeys the promise that its value is always at most polynomial. Immediately from the definitions, FewP $\subseteq$ Few. SPP [OH93,FFK94] is the class of sets $L$ such that for some nondeterministic polynomial-time Turing machine $N$ it holds that if $x \notin L$ then $N(x)$ has one fewer accepting path than it has rejecting paths, and if $x \in L$ then the numbers of accepting and rejecting paths of $N(x)$ are equal. Curiously, note that the nontrivial result that FewP $\subseteq$ SPP itself seems incomparable with the trivial result FewP $\subseteq$ NP.

[3] However, one should keep in mind the contrasting rejection sets of these two classes.

[4] $\mathrm{C}_{=}\mathrm{P}$[half] [BB92] is the class of languages $L$ such that there is some nondeterministic Turing machine such that if the input is in $L$ exactly half of the paths are accepting paths and if the input is not in $L$ none of the paths are accepting paths.

and thus as a corollary to his proof one can claim $(\exists A)\,[\mathrm{C}_{=}\mathrm{P}[\mathrm{half}]^A \not\subseteq \oplus\mathrm{P}^A]$. It follows immediately that $(\exists A)\,[\mathrm{EP}^A \not\subseteq \oplus\mathrm{P}^A]$. Since these are standard diagonalizations that can easily be interleaved, it is easy to see that there is a relativized world in which EP and $\oplus$P are incomparable.[5]

## 4 The Power of Removing EP's Promise

As hinted in footnote 3, EP and $\oplus$P are different sorts of classes. In particular, EP is what is called a promise class. The term "promise class," which was introduced by Hemaspaandra and Rubinstein ([HR92], see also [Rot95,Bor94] for formalizations) to capture a behavior that was shared by a number of extant classes, refers to those classes for which some nontrivial promise—usually regarding number of paths of complementarity of machines—must hold for all inputs. (This notion should not be confused with the different notion of a promise problem [EY80], a notion in which the promise is optional and merely inputs on which the promise holds triggers some secondary requirement.) One particularly central subtype of promise class, which includes those promise classes that we are most concerned with here, is the set of classes that are not currently known to be definable via acceptance and rejection sets that partition $\mathbb{N}$. (For example, though NP can be defined via the rejection set $\{0\}$ and the acceptance set $\{2, 4, 6, \cdots\}$, and these do not partition $\mathbb{N}$, NP does have a scheme, its natural one, that does partition $\mathbb{N}$, and so it is not said to be a promise class.)

Promise classes other than EP include UP, $\mathrm{UP}_{\leq k}$, FewP, Few, NP$\cap$coNP, R, and BPP. Behaviors that are taken for granted for "nice" classes such as NP are often not even known to hold for many promise classes. Among such standard behaviors that promise sets may lack are possession of complete sets (equivalently, "constructive programming systems" [Reg89]) (see, e.g., [Sip82,Gur83,HJV93]), possession of upward separation results ([HJ95], but see [RRW94]), and possession of positive relativization results [HR92]. (See Borchert and Rothe [Rot95,Bor94] for recent thesis-length treatments of promise classes.)

As noted above, EP is a promise class. However, it has a natural non-promise analog.

[5] Concerning $\mathrm{Mod}_q\mathrm{P}$ classes for values $q > 2$ [CH90,BGH90], it follows easily from the known relations among such classes [BGH90] and the obvious fact that powers of 2 are never congruent to zero modulo $j$, where $j$ is any number greater than 2 that is not a power of two, that EP is contained in $\mathrm{Mod}_q\mathrm{P}$ for all $q > 2$ such that $q$ is not a power of two. ($q$ values that are powers of two give just $\oplus$P and thus as noted above are incomparable to EP in some relativized world.) So it also follows from this, in light of Beigel's [Bei91] result that for any distinct primes $q_1$ and $q_2$ there are oracles relative to which $\mathrm{Mod}_{q_1}\mathrm{P}$ and $\mathrm{Mod}_{q_2}\mathrm{P}$ are incomparable, that for every $q > 1$ there is an oracle such that $\mathrm{Mod}_q\mathrm{P}$ is not contained in EP.

We will call this analog ES. ES is the class that shares EP's acceptance set but that removes its promise (that is, its promise never to have 3 accepting paths or 5 accepting paths or etc.).

**Definition 4.1** ES *denotes the class of all languages $L$ for which there is a nondeterministic polynomial-time Turing machine $N$ such that, for each input $x \in \Sigma^*$, $x \in L \iff \#acc_N(x) \in \{2^i \mid i \in \mathbb{N}\}$.*

In the case of the promise class UP, removing that class's promise is known to boost the class from being a subset of NP to being hard for UP $\cup$ coNP. Theorem 4.2 shows that removing EP's promise also yields a jump in power. ES coincides with the powerful counting class C$_=$P.

**Theorem 4.2** ES = C$_=$P.

**Proof.** ES $\subseteq \{L \mid L \leq_d^p \text{C}_=\text{P}\}$ is immediately clear from the definitions, where $\leq_d^p$ is polynomial-time disjunctive reducibility [LLS75]. So ES $\subseteq$ C$_=$P, as it is known that C$_=$P = $\{L \mid L \leq_d^p \text{C}_=\text{P}\}$ (this result is implicit in the technique of [GNW90], as noted and generalized in [Rot93]; the result also has been obtained in [BCO93]). To show C$_=$P $\subseteq$ ES, consider a C$_=$P machine and the function $f$ giving the number of paths on which it would accept. Let $w(x)$ be the smallest integer such that $2^{w(x)} > f(x)$. Consider the EP machine that on input $x$ has $2^{1+w(x)} - f(x)$ paths that immediately accept, and that also has paths that simulate the C$_=$P machine. Note that this machine accepts the C$_=$P language. ∎

As C$_=$P is quite powerful—PH $\subseteq$ BP$\cdot$C$_=$P [TO92] and coNP $\subseteq$ C$_=$P—ES is also powerful. For example, it is clear that ES $\subseteq$ NP collapses the polynomial hierarchy to NP $\cap$ coNP.

So, in light of Theorem 4.2, we can say that EP is a promise analog of C$_=$P. The reason we say "*a* promise analog" is that if one builds a promise version of C$_=$P by simply altering the definition of C$_=$P directly via adding the natural promise, one would get ListP$_1$ (formally, ListP$_{\{\lambda n.1\}}$), in the notation of the forthcoming Definition 5.1. Though clearly UP $\subseteq$ ListP$_1$, it is far from clear whether EP = ListP$_1$; that is, these two promise analogs of C$_=$P may differ. In fact, they may differ pairwise not only with each other but with a third promise analog. Namely, C$_{=}$P[half] is another class that can be viewed as a promise analog of C$_=$P, due to the well-known fact that C$_=$P has, as one of its normal forms, the class all sets $L$ such that for some nondeterministic polynomial-time Turing machine $N$ it holds that $L$ is the set of inputs on which exactly half of $N$'s computation paths accept. Clearly C$_{=}$P[half] $\subseteq$ EP $\cap$ ListP$_1$, but it is an open question whether C$_{=}$P[half] equals either EP or ListP$_1$.

## 5 Open Questions

Does EP equal NP? It would be nice to give evidence that such an equality would, for example, collapse the polynomial hierarchy. However, UP $\subseteq$ EP $\subseteq$ NP, and at the present time, it is open whether even the stronger assumption UP = NP implies any startling collapses. Also regarding the placement of EP, can one show that EP is contained in SPP, or can one show that EP is contained in the larger counting classes WPP or LWPP [FFK94]? Also, does EP, in contrast to most promise classes, have complete sets? We conjecture that EP lacks complete sets (of course, if EP equals NP then EP has complete sets).

It is clear that EP is closed under conjunctive reductions and under disjoint union, and (thus) under intersection. Is EP closed under disjunctive reductions or union?

Finally, define:

**Definition 5.1** $\mathrm{ListP}_{\mathcal{F}}$ *is the class of all sets* $L$ *such that* $(\exists f \in \mathrm{FP},\ f : \Sigma^* \to 2^{\mathbb{N}})$ $(\exists h \in \mathcal{F})$ $(\exists$ *nondeterministic polynomial-time Turing machine* $N)$ $(\forall x)$

$$[||f(x)|| \leq h(|x|) \wedge \#acc_N(x) \in f(x) \cup \{0\} \wedge (x \in L \iff \#acc_N(x) \geq 1)],$$

*where* FP *denotes the class of (total) polynomial-time functions.*

Let $\mathrm{ListP}_{\mathrm{poly}}$ = ListP, where poly is the set of all polynomials. That is, ListP is very similar to EP, except the list of potential numbers of accepting paths on an input, rather than being $\{0, 1, 2, 4, 8, \cdots\}$, is instead some polynomial-time computable polynomial-sized list (of numbers written in binary) that may depend on the input. Clearly, EP $\subseteq$ ListP, and in fact the EP analogs based not on powers of 2 but on powers of $k$ are also in ListP. Is SAT $\in$ ListP? We do not know. However, note that ListP is a language cousin of the "function" notion of enumerative counting introduced by Cai and Hemaspaandra [CH89]. It follows immediately from a (later) result of Cai and Hemaspaandra [CH91] and, independently, Amir, Beigel, Gasarch, and Toda (as cited in [ABG90]) that SAT is in ListP *via a machine* $M$ *that uses the canonical witness scheme for* SAT *(or any witness scheme whose numbers of witnesses are 1-Turing interreducible with those of the canonical witness scheme in the context of the input)* if and only if $\mathrm{P} = \mathrm{P}^{\#\mathrm{P}}$. So if we knew that all witness schemes for SAT were closely related to the canonical one, then we would know that SAT was in ListP if and only if $\mathrm{P} = \mathrm{P}^{\#\mathrm{P}}$. However, can SAT thwart this by having some bizarre witness scheme deeply unrelated to its canonical witness scheme? In fact, Fischer, Hemaspaandra, and Torenvliet have recently provided sufficient conditions for such schemes to exist [FHT95].

Similar comments apply to Graph Automorphism (GA). Is GA $\in$ ListP? We do not know. However, it follows immediately from a result of Chang, Gasarch, and Torán [CGT95] that GA is in ListP *via a machine M that uses the canonical witness scheme for GA (or any witness scheme whose numbers of witnesses are 1-Turing interreducible with those of the canonical witness scheme in the context of the input)* if and only if Graph *Isomorphism* is in R, random polynomial time [Gil77].

**Acknowledgments**

We thank Dieter Kratsch, Haiko Müller, and Johannes Waldmann for very kindly letting us use their office's computers to type in this paper. We thank Richard Beigel for helpful proofreading comments on an earlier version.

## References

[ABG90] A. Amir, R. Beigel, and W. Gasarch. Some connections between bounded query classes and non-uniform complexity. In *Proceedings of the 5th Structure in Complexity Theory Conference*, pages 232–243. IEEE Computer Society Press, July 1990.

[All86] E. Allender. The complexity of sparse sets in P. In *Proceedings of the 1st Structure in Complexity Theory Conference*, pages 1–11. Springer-Verlag *Lecture Notes in Computer Science #223*, June 1986.

[AR88] E. Allender and R. Rubinstein. P-printable sets. *SIAM Journal on Computing*, 17(6):1193–1202, 1988.

[AT96] M. Agrawal and T. Thierauf. The boolean isomorphism problem. Technical Report TR 96-032, Electronic Colloquium on Computational Complexity, March 1996.

[BB92] A. Berthiaume and G. Brassard. The quantum challenge to structural complexity theory. In *Proceedings of the 7th Structure in Complexity Theory Conference*, pages 132–137. IEEE Computer Society Press, June 1992.

[BCO93] R. Beigel, R. Chang, and M. Ogiwara. A relationship between difference hierarchies and relativized polynomial hierarchies. *Mathematical Systems Theory*, 26:293–310, 1993.

[Bei89] R. Beigel. On the relativized power of additional accepting paths. In *Proceedings of the 4th Structure in Complexity Theory Conference*, pages 216–224. IEEE Computer Society Press, June 1989.

[Bei91] R. Beigel. Relativized counting classes: Relations among thresholds, parity, and mods. *Journal of Computer and System Sciences*, 42(1):76–96, 1991.

[BG82] A. Blass and Y. Gurevich. On the unique satisfiability problem. *Information and Control*, 55:80–88, 1982.

[BGH90] R. Beigel, J. Gill, and U. Hertrampf. Counting classes: Thresholds, parity, mods, and fewness. In *Proceedings of the 7th Annual Symposium on Theoretical Aspects of Computer Science*, pages 49–57. Springer-Verlag *Lecture Notes in Computer Science #415*, February 1990.

[Bor94] B. Borchert. *Predicate Classes, Promise Classes, and the Acceptance Power of Regular Languages*. PhD thesis, Universität Heidelberg, Mathematisches Institut, Heidelberg, Germany, 1994.

[BRS95] B. Borchert, D. Ranjan, and F. Stephan. On the computational complexity of some classical equivalence relations on boolean functions. Technical Report TR 18, Universität Heidelberg, Mathematisches Institut, Heidelberg, Germany, December 1995.

[Bry92] R. Bryant. Symbolic boolean manipulation with ordered binary decision diagrams. *ACM Computing Surveys*, 24(3):293–318, 1992.

[CGT95] R. Chang, W. Gasarch, and J. Torán. On finding the number of graph automorphisms. In *Proceedings of the 10th Structure in Complexity Theory Conference*, pages 288–298. IEEE Computer Society Press, June 1995.

[CH89] J. Cai and L. Hemachandra. Enumerative counting is hard. *Information and Computation*, 82(1):34–44, 1989.

[CH90] J. Cai and L. Hemachandra. On the power of parity polynomial time. *Mathematical Systems Theory*, 23(2):95–106, 1990.

[CH91] J. Cai and L. Hemachandra. A note on enumerative counting. *Information Processing Letters*, 38(4):215–219, 1991.

[EY80] S. Even and Y. Yacobi. Cryptocomplexity and NP-completeness. In *Proceedings of the 7th International Colloquium on Automata, Languages, and Programming*, pages 195–207. Springer-Verlag *Lecture Notes in Computer Science*, 1980.

[FFK94] S. Fenner, L. Fortnow, and S. Kurtz. Gap-definable counting classes. *Journal of Computer and System Sciences*, 48(1):116–148, 1994.

[FHS78] S. Fortune, J. Hopcroft, and E. Schmidt. The complexity of equivalence and containment for free single program schemes. In *Proceedings of the 5th International Colloquium on Automata, Languages, and Programming*, pages 227–240. Springer-Verlag *Lecture Notes in Computer Science #62*, 1978.

[FHT95] S. Fischer, L. Hemaspaandra, and L. Torenvliet. Witness-isomorphic reductions and the local search problem. In *Proceedings of the 20th Symposium on Mathematical Foundations of Computer Science*, pages 277–287. Springer-Verlag *Lecture Notes in Computer Science #969*, August/September 1995.

[FK92] M. Fellows and N. Koblitz. Self-witnessing polynomial-time complexity and prime factorization. In *Proceedings of the 7th Structure in Complexity Theory Conference*, pages 107–110. IEEE Computer Society Press, June 1992.

[Gil77] J. Gill. Computational complexity of probabilistic Turing machines. *SIAM Journal on Computing*, 6(4):675–695, 1977.

[GNW90] T. Gundermann, N. Nasser, and G. Wechsung. A survey on counting classes. In *Proceedings of the 5th Structure in Complexity Theory Conference*, pages 140–153. IEEE Computer Society Press, July 1990.

[GP86] L. Goldschlager and I. Parberry. On the construction of parallel computers from various bases of boolean functions. *Theoretical Computer Science*, 43:43–58, 1986.

[GS88] J. Grollmann and A. Selman. Complexity measures for public-key cryptosystems. *SIAM Journal on Computing*, 17(2):309–335, 1988.

[Gur83] Y. Gurevich. Algebras of feasible functions. In *Proceedings of the 24th IEEE Symposium on Foundations of Computer Science*, pages 210–214. IEEE Computer Society Press, November 1983.

[Har71] M. Harrison. Counting theorems and their applications to classification of switching functions. In A. Mukhopadyay, editor, *Recent Developments in Switching Theory*, pages 4–22. Academic Press, 1971.

[HH94] E. Hemaspaandra and L. Hemaspaandra. Quasi-injective reductions. *Theoretical Computer Science*, 123:407–413, 1994.

[HJ95] L. Hemaspaandra and S. Jha. Defying upward and downward separation. *Information and Computation*, 121(1):1–13, 1995.

[HJV93] L. Hemaspaandra, S. Jain, and N. Vereshchagin. Banishing robust Turing completeness. *International Journal of Foundations of Computer Science*, 4(3):245–265, 1993.

[HR] L. Hemaspaandra and J. Rothe. Unambiguous computation: Boolean hierarchies and sparse Turing-complete sets. *SIAM Journal on Computing*. To appear.

[HR92] L. Hemachandra and R. Rubinstein. Separating complexity classes with tally oracles. *Theoretical Computer Science*, 92(2):309–318, 1992.

[Ko85] K. Ko. On some natural complete operators. *Theoretical Computer Science*, 37:1–30, 1985.

[KSTT92] J. Köbler, U. Schöning, S. Toda, and J. Torán. Turing machines with few accepting computations and low sets for PP. *Journal of Computer and System Sciences*, 44(2):272–286, 1992.

[LLS75] R. Ladner, N. Lynch, and A. Selman. A comparison of polynomial time reducibilities. *Theoretical Computer Science*, 1(2):103–124, 1975.

[OH93] M. Ogiwara and L. Hemachandra. A complexity theory for closure properties. *Journal of Computer and System Sciences*, 46:295–325, 1993.

[PZ83] C. Papadimitriou and S. Zachos. Two remarks on the power of counting. In *Proceedings 6th GI Conference on Theoretical Computer Science*, pages 269–276. Springer-Verlag *Lecture Notes in Computer Science #145*, 1983.

[Reg89] K. Regan. Provable complexity properties and constructive reasoning. Manuscript, April 1989.

[Rot93] J. Rothe. Some closure properties of GAP-definable classes. Technical Report TR 6-93, Institut für Informatik, Friedrich-Schiller-Universität Jena, Jena, Germany, 1993.

[Rot95] J. Rothe. *On Some Promise Classes in Structural Complexity Theory.* PhD thesis, Institut für Informatik, Friedrich-Schiller-Universität Jena, Jena, Germany, 1995.

[RRW94] R. Rao, J. Rothe, and O. Watanabe. Upward separation for FewP and related classes. *Information Processing Letters*, 52(4):175–180, 1994.

[Sch89] U. Schöning. Probabilistic complexity classes and lowness. *Journal of Computer and System Sciences*, 39(1):84–100, 1989.

[Sim75] J. Simon. *On Some Central Problems in Computational Complexity.* PhD thesis, Cornell University, Ithaca, N.Y., January 1975. Available as Cornell Department of Computer Science Technical Report TR75-224.

[Sip82] M. Sipser. On relativization and the existence of complete sets. In *Proceedings of the 9th International Colloquium on Automata, Languages, and Programming*, pages 523–531. Springer-Verlag *Lecture Notes in Computer Science #140*, 1982.

[Smi84] L. Smith. *Linear Algebra.* Springer-Verlag, 2nd edition, 1984.

[TO92] S. Toda and M. Ogiwara. Counting classes are at least as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 21(2):316–328, 1992.

[Val76] L. Valiant. The relative complexity of checking and evaluating. *Information Processing Letters*, 5:20–23, 1976.

[Ver94] N. Vereshchagin. Relativizable and nonrelativizable theorems in the polynomial theory of algorithms. *Russian Academy of Sciences–Izvestiya–Mathematics*, 42(2):261–298, 1994.

[Wag86] K. Wagner. The complexity of combinatorial problems with succinct input representations. *Acta Informatica*, 23:325–356, 1986.

[Wat88] O. Watanabe. On hardness of one-way functions. *Information Processing Letters*, 27:151–157, 1988.