Open Problems Column
Edited by William Gasarch

This Issue’s Column!
This issue’s Open Problem Column is by Lance Fortnow and its titled Worlds to Die For: Open Oracle Questions for the 21st Century.

Request for Columns!
I invite any reader who has knowledge of some area to contact me and arrange to write a column about open problems in that area. That area can be (1) broad or narrow or anywhere inbetween, and (2) really important or really unimportant or anywhere inbetween.
Abstract

Most of the interesting open problems about relationships between complexity classes have either been resolved or have relativizable worlds in both directions. We discuss some remaining open questions, updating questions from a similar 1995 survey of Hemaspaandra, Ramachandra and Zimand and adding a few new problems.

1 Introduction

Most of the interesting questions in complexity have relativized worlds in both directions, like whether the polynomial-time hierarchy is infinite [Yao85, Hås89], or whether the Berman-Hartmanis isomorphism conjecture holds [FFK96, BBF98].

Some have questioned the importance of relativization after the non-relativizing techniques of interactive proof systems [FS88, LFKN92]. However these techniques have had limited use and in the thirty years hence we have not seen any significant new non-relativizing techniques and there have been no new examples of theorems that go against previously published relativization results.

For an introduction and definitions of relativization I recommend the original oracle paper by Baker, Gill and Solovay [BGS75], the author’s survey on the role of relativization in complexity [For94] and

In 1995, Lane Hemaspaandra, Ajit Ramachandrand and Marius Zimand wrote a survey entitled “Worlds to Die For” [HRZ95], giving a list of several open questions about the existence of oracles and random oracles that would make various complexity classes true. We review progress in these questions in Section 5. Even though 26 years have passed, only a few of the problems mentioned by Hemaspaandra et al. have been fully solved showing the vast difficulty of these problems.

2 The Complexity Zoo

The Complexity Zoo [Aar], created by Scott Aaronson, lists most, if not all, of the known complexity classes. For the many classes mentioned in this survey I recommend visiting the zoo if you are unfamiliar with them.
One can use the zoo itself for finding open questions, looking for the minimal classes \( C \) and \( D \) where it is open whether \( C \subseteq D \) and whether there is an oracle \( A \) such that \( C^A \not\subseteq D^A \).

Robert Sanders [San19] automated this process to generate open questions, such as whether \( \text{BQP} \) has (non-quantum) interactive proofs. Since \( \text{BQP} \) is in \( \text{PSPACE} \) [ADH97], languages in \( \text{BQP} \) have interactive proof systems in the “real world” making the oracle question less interesting. Automating the creation of open questions might reveal some good problems otherwise missed but one needs to carefully check that the problems would reveal new insights into complexity.

3 How to Solve Tough Oracle Questions

Underlying many oracle questions is a combinatorics one. In the most famous example, Yao’s paper “Separating the polynomial-time hierarchy by oracle” [Yao85] is heralded more for showing parity doesn’t have constant-depth polynomial-size circuits. Some of the oracle questions below, specifically Problems 5.5 and 6.4, are hard because the underlying combinatorial problems are difficult.

Another challenge for oracle constructions are trying to fulfill conflicting requirements. For example in Problem 5.4 we know how to collapse the Boolean hierarchy or make the polynomial-time hierarchy infinite with relativizations but doing both at the same time conflict in ways we don’t know how to resolve.

One approach is to create the oracle first and then show it has the right properties. Fenner, Fortnow and Kurtz [FFK94] used this approach to give the first oracle relative to which the Berman-Hartmanis isomorphism conjecture held. The “An Oracle Builders Toolkit” [FFKL03] by Steve Fenner, Lance Fortnow, Stuart Kurtz and Lide Li, explores this approach in detail.

The questions in this survey have stymied the various attempts to settle them and likely some clever new techniques will be needed.

4 The Extremes

These are oracles which push the limits of what we don’t know how to prove and of which many other oracle results follow. Pushing these limits further are good open problems.

4.1 \( P = \text{PSPACE} \)

Discovered by Baker, Gill and Solovay in the original oracle paper [BCS75].

Collapses everything between \( P \) and \( \text{PSPACE} \).

\[ P = \text{NP} = \text{coNP} = \text{PH} = \oplus P = PP = \text{BQP} = P^{\#P} = \text{PSPACE} = \text{NPSPACE} \]
4.2 Generic Oracles

First applied to computational complexity by Blum and Impagliazzo [BI87].

These oracles separate as much as possible. $P \neq NP$, the polynomial-time hierarchy is infinite, $PP$ is not in $P^{NP}$, $NP$ is not in $\oplus P$ and much more. Oddly enough they diagonalize against machines that need to fulfill a promise condition and so with the appropriate construction one also gets $P = NP \cap coNP = UP = BPP = BQP$.

4.3 $P = \oplus P$ and $NP = EXP$

First proved by Beigel, Buhrman and Fortnow [BBF98].

Relative to this oracle $ZPP = \oplus EXP$ and the Berman-Hartmanis isomorphism conjecture [BH77] holds (all $NP$-complete problems are reducible to each other via invertible bijections).

4.4 $P = NP$ and $\oplus P = EXP$

Discovered by Beigel and Maciel [BM99].

The polynomial-time hierarchy collapses to $P$ and yet the exponential hierarchy sits inside $\oplus P$.

4.5 $P = \oplus P$ and $BPP = EXP^{NP}$

Discovered by Buhrman and Torenvliet [HT00, Corollary 4.8].

No even very weak derandomization for $BPP$. Valiant-Vazirani puts $NP$ in $RP^{\oplus P}$ but in this oracle $NP$ is not even in $coNP^{\oplus P}$.

4.6 $P^{RP} = NEXP$

Discovered by Buhrman, Fenner, Fortnow and Torenvliet [BFFT01].

No even very weak derandomization for $RP$. Implies $P^{NP} = P^{NEXP}$ (also implied by the next oracle).

4.7 $P^{NP} = \oplus P = P^{PEXP}$

A strong version of Beigel’s oracle where $P^{NP}$ is not in $PP$ [Bei91] (though the entire polynomial-time is in $P^{PP}$ relative to any oracle [Tod91]) and $PP$ is not closed under Turing-reductions.

You can replace $\oplus P$ with $\text{Mod}_k P$ for any prime $k$ in any of the above. We don’t believe any of the statements to be true in the unrelativized world but all of them remain open and would require nonrelativizing techniques to disprove.
5 Updates to Die For

A brief update on the problems asked by Hemaspaandra et al. [HRZ95]. We refer you back to that paper for details on the problems.

5.1 Show that with probability one, the polynomial time hierarchy is proper

Resolved by Håstad, Rossman, Servedio and Tan [HRST17] who showed that a random oracle will, with probability one, separate all the levels of the polynomial-time hierarchy.

5.2 Construct an oracle relative to which BPTIME[\(n\)] = BPTIME[\(n^2\)].

Still open in its full generalization. Rettinger and Verbeek [RV01] show there exists an oracle \(A\) such that BPP\(^A\) = BPTIME\(^A\)(\(n\)), equality if the machines only have truth-table (non-adaptive) access to the oracle.

You can get a separation (for all oracles) if you allow the machine a single bit of advice, i.e., BPTIME(\(n^2\))/1 \(\neq\) BPTIME(\(n\))/1 (see [FS06]).

5.3 Show that relative to a random oracle there are secure pseudorandom generators

Settled in the affirmative by Marius Zimand [Zim98].

5.4 Build an oracle such that the Boolean Hierarchy collapses yet differs from the polynomial hierarchy

Remains open. We can show that if \(P^{NP[1]} = P^{NP[2]}\) then \(P = P^{NP}\) [BF99] and \(PH = S^p_2 \subseteq ZPP^{NP}\) [FPS08] and these proofs relativize.

Nevertheless there still may be relativized world where \(P^{NP[1]} = P^{NP[2]}\) and \(P^{NP} \neq ZPP^{NP}\). To settle 5.4 in the negative, one would need relativizable proofs of the following.

1. Prove the same consequence for any collapse of the Boolean hierarchy, i.e., \(P^{NP[k]} = P^{NP[k+1]}\) for some \(k\) implies \(P = P^{NP}\) and \(PH = S^p_2 \subseteq ZPP^{NP}\). I suspect this would follow with a careful analysis of the earlier proofs.

2. Get the final collapse of \(ZPP^{NP}\) to \(P^{NP}\). This would seem to require a new technique.
5.5 Build an oracle relative to which $\text{PP}^{\oplus P} \subsetneq \text{PSPACE}$.  
Still open. This question is roughly equivalent to showing that any polylogarithmic degree polynomial over $\text{GF}[2]$ has exponentially-small correlation with $\text{Mod}_3$, or some other problem in $\text{NC}^1$. Jean Bourgain [Bou05] shows exponentially-small correlation for degree slightly less than log. Bhowmick and Lovett [BL15] proposed a potential barrier that could explain the difficulty.

5.6 Build a tally oracle $T$ such that $P^T \neq \text{NP}^T$.  
This one was a bit of a joke in the Hemaspaandra et al. survey as it is equivalent to $P \neq \text{NP}$ unrelativized. Still open.

5.7 Build an oracle $A$ relative to which $\text{SPP}$ has no complete sets.
Resolved in the affirmative. Vereschagin [Ver93] showed that, modulo some technical restrictions, that if there was an oracle making a class $C$ different than $P$ and generic oracles collapse $C$ and $P$ then there is an oracle where $C$ does not have complete sets. Fenner, Fortnow, Kurtz and Li [FFKL03] showed these conditions held for $\text{SPP}$.

6 A Few of My Favorite Open Oracle Questions
I’ve banged my head against the wall on all of these problems. Please help out.

6.1 Build an oracle where $\text{UP} = \text{NP}$ and the polynomial-time hierarchy is infinite
Suppose for every $\text{NP}$ machine $M$ there is another $\text{NP}$ machine $N$ such that for all $x$ if $M(x)$ has an accepting path then

1. $N(x)$ has a unique accepting path
2. The accepting path of $N(x)$ encodes a single accepting path of $M(x)$.

Hemaspaandra, Naik, Ogihara and Selman [HNOS96] show that under this condition the polynomial-time hierarchy collapses to the second level. $\text{NP} = \text{UP}$ is similar but doesn’t require condition (2), the $\text{UP}$ machine’s accepting path may be unrelated to any accepting path of the $\text{NP}$ machine. We don’t know if $\text{NP} = \text{UP}$ implies the polynomial-time hierarchy collapses, or a relativized counterexample.
6.2 Build an oracle that separates $S^p_2$ from $ZPP^{NP}$

Jin-Yi Cai [Cai07] has a clever proof that $S^p_2 \subseteq ZPP^{NP}$ and the original $S^p_2$ paper [RS98] shows $P^{NP} \subseteq S^p_2$. Russell and Sundaram [RS98] also give a relativizable proof that $BPP \subseteq S^p_2$ which implies an oracle separating $P^{NP}$ from $S^p_2$ (follows from 4.5). Whether there is a relativizing proof showing that $ZPP^{NP} \subseteq S^p_2$ remains an interesting open problem.

6.3 Build an oracle where the polynomial-time hierarchy looks like the arithmetic hierarchy

We know the structure of the arithmetic hierarchy in computability theory: for all $k$, $\Sigma^k_2 \neq \Sigma^{k+1}_2$ and $\Sigma^{k+1}_2 \cap \Pi^{k+1}_2$ are exactly the languages computable by a Turing machine with an oracle for $\Sigma_k$.

Can we create an oracle relative to which the polynomial-time hierarchy has the same structure, i.e., the hierarchy is infinite and $\Sigma^{p+1}_k \cap \Pi^{p+1}_k = P^{\Sigma^p_k}$ for all $k \geq 0$?

Relative to a generic oracle, the polynomial-time hierarchy is infinite and $P = NP \cap coNP$ [BI87] which suggested that generic oracles could give the oracle for Problem 6.3. Fortnow and Yamakami [FY96] show this isn’t true, $\Sigma^{p+1}_k \cap \Pi^{p+1}_k \neq P^{\Sigma^p_k}$ for $k \geq 1$.

One can generalize this question in uncountably many ways. Let $\alpha = \alpha_0\alpha_1\ldots$ be an infinite binary string. Build an oracle relative to which

1. The polynomial-time hierarchy is infinite.

2. For all $k$,
   
   (a) If $\alpha_k = 0$ then $\Sigma^{p+1}_k \cap \Pi^{p+1}_k = P^{\Sigma^p_k}$.
   
   (b) If $\alpha_k = 1$ then $\Sigma^{p+1}_k \cap \Pi^{p+1}_k \neq P^{\Sigma^p_k}$.

Problem 6.3 asks to build an oracle for $\alpha = 000\ldots$. The only solved cases are for generic oracles that satisfy the above for $\alpha = 011\ldots$ and by straightforward diagonalization for $\alpha = 111\ldots$. There is no $\alpha$ for which we know no such oracle exists.

On the other hand we can’t even rule out a relativizable proof that $\Sigma^p_2 \cap \Pi^p_2 = P^{NP}$ implies $\Sigma^p_2 = \Pi^p_2$.

6.4 Is $P \neq BQP$ relative to a random oracle?

This is widely believed to be true. If $P = BQP$ relative to a random oracle than $BQP = BPP$ [FR99], and thus the factoring problem has an efficient probabilistic algorithm.

Consider a variation: If $P = \text{PSPACE}$ (unrelativized) then $P = BQP$ relative to a random oracle. This statement follows from Conjecture 6.1 below about decision tree complexity.
Conjecture 6.1 For all $p(x_1, \ldots, x_n)$, polynomials of degree $d$ over the reals such that

1. For all $(x_1, \ldots, x_n) \in \{0, 1\}^n$, $p(x_1, \ldots, x_n) \in [0, 1]$.

2. For 0.99 fraction of the $(x_1, \ldots, x_n) \in \{0, 1\}^n$, $|p(x_1, \ldots, x_n) - 1/2| \geq 1/6$.

There is a function $f : \{0, 1\}^n \to \{0, 1\}$ of decision tree complexity polynomial in $d$ and $\log n$ such that for 0.51 fraction of the $(x_1, \ldots, x_n) \in \{0, 1\}^n$, $|p(x_1, \ldots, x_n) - f(x_1, \ldots, x_n)| \leq 1/3$.

Aaronson and Ambainis [AA14] give some evidence that Conjecture 6.1 might be false.

6.5 Build an oracle where for all $f \in \#P$, $P^f \neq P^{NP}$

A language $L$ is checkable if given a program $P$ as an oracle there is a polynomial-time probabilistic algorithm $A$ such that for all $x$,

1. If $P(x) \neq L(x)$ then $A^P(x)$ outputs “$P$ is wrong on some input”.

2. If for all $y$, $L(y) = P(y)$ then $A^P(x)$ outputs “$P$ is correct on input $x$.”

Blum and Kannan [BK89] show that a language $L$ is checkable if and only if both $L$ and $L$ have multi-prover interactive proof systems where the provers are computable in probabilistic polynomial-time with oracle access to $L$. Early results on interactive proofs imply $\#P$-complete, $\text{PSPACE}$-complete, $\text{EXP}$-complete and Graph Isomorphism are all checkable among others. Whether $\text{NP}$-complete problems are checkable is still a major open question and boils down to Conjecture 6.2.

Conjecture 6.2 There is a multi-prover interactive proof system for $\text{SAT}$ where the provers are probabilistic polynomial-time with oracle access to $\text{SAT}$.

We already know Conjecture 6.2 fails relative to an oracle because Fortnow, Rompel and Sipser [FRS94] give an oracle relative to which $\text{coNP}$ has no multi-prover interactive proof no matter how powerful the prover. But this doesn’t give much evidence against Conjecture 6.2 since Lund, Fortnow, Karloff and Nisan [LFKN92] show that $\text{coNP}$ does have an interactive proof system.

There is a related complexity conjecture that may, or may not, shed light on the checkability question.

Conjecture 6.3 Is there a function $f \in \#P$ such that $P^f = P^{NP}$?

Even an $f$ such that $\text{NP} \subseteq P^f \subseteq \text{PH}$ remains open.

One would hope that if such an $f$ existed one could use it as a basis of an interactive proof for $\text{SAT}$. Unfortunately it’s difficult to find an $f$ in $\#P$ that captures $\text{NP}$ without counting solutions. Trying to find a relativized counterexample seems equally difficult.
Acknowledgments

Thanks to Bill Gasarch who suggested this column, to Lane Hemaspaandra and Marius Zimand for discussions of problems old (Section 5) and new, and to Fred Green for updates on the polynomial questions in Section 5.5. Most of Section 4 first appeared in the author’s blog [For05].

References


