Structural Complexity Theory: Recent Surprises*

Juris Hartmanis
Richard Chang**
Desh Ranjan
Pankaj Rohatgi

TR 90-1117
April 1990

Department of Computer Science
Cornell University
Ithaca, NY 14853-7501

*This research was supported in part by NSF Research Grant CCR-88-23053.
**Supported in part by an IBM Graduate Fellowship.

Structural Complexity Theory: Recent Surprises[†]

Juris Hartmanis *Richard Chang*[‡] *Desh Ranjan* *Pankaj Rohatgi*

Department of Computer Science
Cornell University
Ithaca, New York 14853

Abstract

This paper reviews the impact of some recent results on the research paradigms in structural complexity theory.

> *On a paper submitted by a physicist colleague:*
> *"This isn't right. This isn't even wrong."*
> *—Wolfgang Pauli*

1 Introduction

Computational complexity theory studies the quantitative laws which govern computing. It seeks a comprehensive classification of problems by their intrinsic difficulty and an understanding of what makes these problems hard to compute. The key concept in classifying the computational complexity of problems is the complexity class which consists of all the problems solvable on a given computer model within a given resource bound.

Structural complexity theory is primarily concerned with the relations among various complexity classes and the internal structure of these classes. Figure 1 shows some major complexity classes. Although much is known about the structure of these classes, there have not been any results which separate any of the classes between P and PSPACE. We believe that all these classes are different and regard the problem of proving the exact relationship between these classes as the Grand Challenge of complexity theory.

The awareness of the importance of P, NP, PSPACE, etc, has lead to a broad investigation of these classes and to the use of relativization. Almost all of the major results in recursive function theory also hold in relativized worlds. Quite the contrary happens in complexity theory. It was shown in 1975 [BGS75] that there exist oracles A and B such that

$$\mathrm{P}^A = \mathrm{NP}^A \quad \text{and} \quad \mathrm{P}^B \neq \mathrm{NP}^B.$$

[†] This research was supported in part by NSF Research Grant CCR 88-23053.

[‡] Supported in part by an IBM Graduate Fellowship.

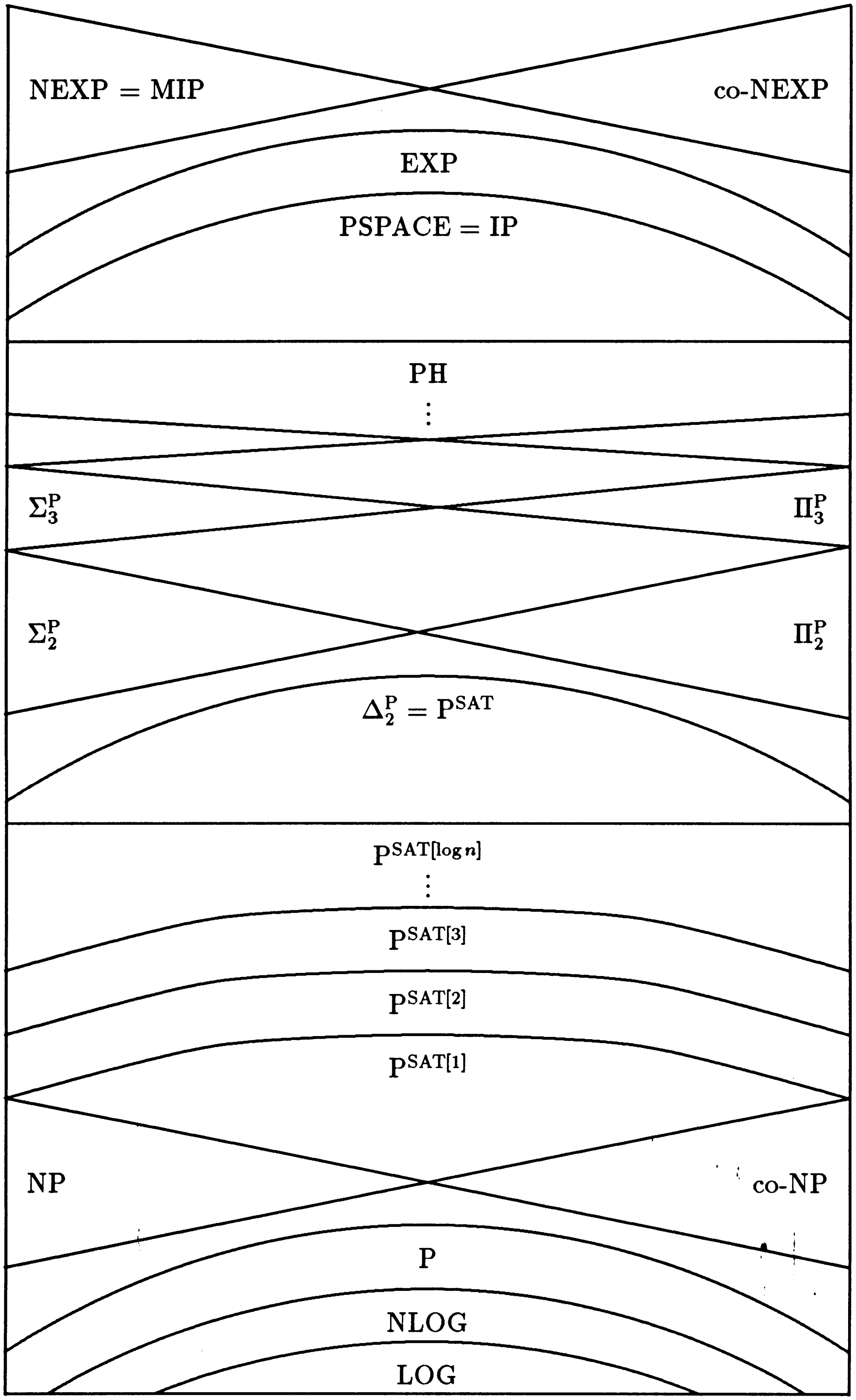

Figure 1: Some standard complexity classes.

This was followed by an extensive investigation of the structure of complexity classes under relativization. An impressive set of techniques was developed for oracle constructions and some very subtle and interesting relativization results were obtained. For example, for a long time it was not known if the Polynomial-time Hierarchy (PH) can be separated by oracles from PSPACE. In 1985, A. Yao [Yao85] finally resolved this problem by constructing an oracle A, such that

$$\mathrm{PH}^A \neq \mathrm{PSPACE}^A.$$

These methods were refined by Ko [Ko88] to show that for every $k \geq 0$ there is an oracle which collapses PH to exactly the k^{th} level and keeps the first $k-1$ levels of PH distinct. That is, for all k, there exists an A such that

$$\Sigma_0^{\mathrm{P},A} \neq \Sigma_1^{\mathrm{P},A} \neq \cdots \neq \Sigma_k^{\mathrm{P},A} \text{ and } \Sigma_k^{\mathrm{P},A} = \Sigma_{k+i}^{\mathrm{P},A},\ i \geq 0.$$

Another aspect of relativized computations was studied by Bennett and Gill who decided to count the number of oracles which separate certain complexity classes. They showed that $\mathrm{P}^A \neq \mathrm{NP}^A$ for almost all oracles. More precisely, they showed that for random oracles the following separations occur with probability 1 [BG81]:

$$\mathrm{Prob}_A[\ \mathrm{P}^A \neq \mathrm{NP}^A \neq \text{co-NP}^A\ ] = 1$$
$$\mathrm{Prob}_A[\ \mathrm{SPACE}^A[\log n] \neq \mathrm{P}^A\ ] = 1$$
$$\mathrm{Prob}_A[\ \mathrm{PSPACE}^A \neq \mathrm{EXP}^A\ ] = 1$$
$$\mathrm{Prob}_A[\ \mathrm{P}^A = \mathrm{RP}^A = \mathrm{BPP}^A\ ] = 1.$$

Many other interesting probability 1 results followed [Cai86,Cai87,KMR89]:

$$\mathrm{Prob}_A[\ \mathrm{BH}^A \text{ is infinite }] = 1$$
$$\mathrm{Prob}_A[\ \mathrm{PH}^A \subsetneq \mathrm{PSPACE}^A\ ] = 1$$
$$\mathrm{Prob}_A[\text{ Under relativization the Berman-Hartmanis Conjecture fails }] = 1.$$

The last result asserts that there exist non-isomorphic many-one complete sets for NP^A with probability 1. It was conjectured that all NP many-one complete sets are polynomial-time isomorphic [BH77].

Surveying the rich set of relativization results, we can make several observations. First, almost all questions about the relationship between the major complexity classes have contradictory relativizations. That is, there exist oracles which separate the classes and oracles which collapse them. Furthermore, many of our proof techniques relativize and cannot resolve problems with contradictory relativizations. Finally, we have unsuccessfully struggled for over twenty years to resolve whether P =? NP =? PSPACE.

These observations seemed to support the conviction that problems with contradictory relativizations are extremely difficult and may not be solvable by current techniques. This opinion was succinctly expressed by John Hopcroft [Hop84]:

> This perplexing state of affairs is obviously unsatisfactory as it stands. No problem that has been relativized in two conflicting ways has yet been solved, and this fact is generally taken as evidence that the solutions of such problems are beyond the current state of mathematics.

How should complexity theorists remedy "this perplexing state of affairs"? In one approach, we assume as a working hypothesis that PH has infinitely many levels. Thus, any assumption which would imply that PH is finite is deemed incorrect. For example, Karp, Lipton and Sipser [KL80] showed that if $\mathrm{NP} \subseteq \mathrm{P/poly}$, then PH collapses to Σ_2^{P}. So, we believe that SAT does not have polynomial sized circuits. Similarly, we believe that the Turing-complete and many-one complete sets for NP are not sparse, because Mahaney [Mah82] showed that these conditions would collapse PH. One can even show that for any $k \geq 0$, $\mathrm{P}^{\mathrm{SAT}[k]} = \mathrm{P}^{\mathrm{SAT}[k+1]}$ implies that PH is finite [Kad88]. Hence, we believe that $\mathrm{P}^{\mathrm{SAT}[k]} \neq \mathrm{P}^{\mathrm{SAT}[k+1]}$ for all $k \geq 0$. Thus, if the Polynomial Hierarchy is indeed infinite, we can describe many aspects of the computational complexity of NP.

A second approach uses random oracles. Since the probability 1 relativization results agree with what complexity theorists believe to be true in the base case and since random oracles have no particular structure of their own, it seemed that the probability 1 behavior of complexity classes should be the same as the base case behavior. This lead Bennett and Gill to postulate the Random Oracle Hypothesis [BG81] which essentially states that whatever holds with probability 1 for relativized complexity classes holds in the unrelativized case—i.e., the real world.

In the following, we will discuss a set of results about interactive proofs which provide dramatic counterexamples to the belief that problems with contradictory relativizations cannot be resolved with known techniques. They also add a striking new counterexample against the Random Oracle Hypothesis. There have been other counterexamples in the literature [Kur82,Har85], but none as compelling. Thus, contradictory relativizations should no longer be viewed as strong evidence that a problem is beyond our grasp. We hope that these results will encourage complexity theorists to renew the attack on problems with contradictory relativizations.

2 A Review of IP

The class IP is the set of languages that have interactive proofs or protocols. IP was first defined as way to generalize NP. NP can be characterized as being precisely those languages for which one can present a polynomially long proof to certify that the input string is in the language. Moreover, the proof can be checked in polynomial time. It is this idea of presenting and checking the proof that the definition of IP generalizes.

Is there a way of giving convincing evidence that the input string is in a language without showing the whole proof to a verifier? Clearly, if we do not give a complete proof to a verifier which does not have the power or the time to generate and check a proof, then we cannot expect the verifier to be completely convinced. This leads us to a very fascinating problem: *how can the verifier be convinced with high probability that there is a proof? and how rapidly can this be done?*

This problem has been formulated and extensively studied in terms of *interactive protocols* [Gol89]. Informally, an interactive protocol consists of a *Prover* and a *Verifier*. The Prover is an all powerful Turing Machine (TM) and the Verifier is a TM which operates in time polynomial in the length of the input. In addition, the Verifier has a random source (e.g., a fair coin) not visible to the Prover. In the beginning of the interactive protocol the Prover and the Verifier receive the same input string. Then, the Prover tries to convince the Verifier, through a series of queries and answers, that the input string belongs to a given language. The Prover succeeds if the Verifier accepts with probability greater than 2/3. The probability is computed over all possible coin tosses made by the Verifier. However, the Verifier must guard against imposters masquerading as the real Prover. The Verifier must not be convinced to accept a string not in the language with probability greater than 1/3—even if the Prover lies.

Definition IP: Let V be a probabilistic polynomial time TM and let P be an arbitrary TM. P and V share the same input tape and communicate via a communication tape. P and V forms an interactive protocol for a language L if

1. $x \in L \Longrightarrow \text{Prob}[\ P\text{-}V \text{ accepts } x\ ] > \frac{2}{3}$.
2. $x \notin L \Longrightarrow \forall P^*, \text{Prob}[\ P^*\text{-}V \text{ accepts } x\ ] < \frac{1}{3}$.

A language L is in IP if there exist P and V which form an interactive protocol for L.

Clearly, IP contains all NP languages, because in polynomial time the Prover can give the Verifier the entire proof. In such a protocol, the Verifier cannot be fooled and never accepts a string not in the language. To illustrate how randomness can generalize the concept of a proof, we look at an interactive protocol for a language not known to be in NP. Consider GNI, the set of pairs of graphs that are not isomorphic. GNI is known to be in co-NP and it is believed not to be in NP. However, GNI does have an interactive protocol [GMW86]. For small graphs, the Verifier can easily determine if the two graphs are not isomorphic. For sufficiently large graphs, the Verifier solicits help from the Prover to show that G_i and G_j are not isomorphic, as follows:

1. The Verifier randomly selects G_i or G_j and a random permutation of the selected graph. This process is independently repeated n times, where n is the number of vertices in G_j. If the graphs do not have the same number of vertices, they are clearly not isomorphic. This sequence of n randomly chosen, randomly permuted graphs is sent to the Prover. Recall that the Prover has not seen the Verifier's random bits. This assumption is not necessary, but simplifies the exposition.

2. The Verifier asks the Prover to determine, for each graph in the sequence, which graph, G_i or G_j, was the one selected. If the Prover answers correctly, then the Verifier accepts.

Suppose the two original graphs are not isomorphic. Then, only one of the original graphs is isomorphic to the permuted graph. The Prover simply answers by picking that graph. If the graphs are isomorphic, then the Prover has at best a 2^{-n} chance of answering all n questions correctly. Thus, the Verifier cannot be fooled often. Therefore, GNI $\in$ IP.

Note that GNI is not known to be complete for co-NP. So, the preceding discussion does not show that co-NP $\subseteq$ IP. For a while, it was believed that co-NP is not contained in IP, because there are oracle worlds where co-NP $\not\subseteq$ IP [FS88]. In fact, the computational power of interactive protocols was not fully appreciated until Lund, Fortnow, Karloff and Nisan [LFKN89] showed that IP actually contains the entire Polynomial Hierarchy. This result then led Shamir [Sha89] to completely characterize IP by showing that

$$\mathrm{IP} = \mathrm{PSPACE}.$$

Very recently, Babai, Fortnow and Lund [BFL90] characterized the computational power of multi-prover interactive protocols

$$\mathrm{MIP} = \mathrm{NEXP}.$$

In both cases, it is interesting to see that interactive proof systems provide alternative definitions of classic complexity classes. Thus, they fit very nicely with the overall classification of feasible computations. Furthermore, both of these problems have contradictory relativizations [FS88]. That is, there exist oracles A and B such that

$$\mathrm{IP}^{\mathrm{A}} = \mathrm{PSPACE}^{A} \quad \text{and} \quad \mathrm{IP}^{\mathrm{B}} \neq \mathrm{PSPACE}^{B},$$

and similarly for the multi-prover case. Thus, these results provide the first *natural* counterexamples to the belief that problems with contradictory relativizations are beyond our proof techniques.

3 The Random Oracle Hypothesis ...

In this section we observe that the proof of IP = PSPACE does not relativize and show that for almost all oracles the two relativized classes differ:

$$\mathrm{Prob}_A[\ \mathrm{IP}^{\mathrm{A}} \neq \mathrm{PSPACE}^{A}\ ] = 1.$$

It is easily seen that

$$\mathrm{IP}^{\mathrm{PSPACE}} = \mathrm{PSPACE}^{\mathrm{PSPACE}}$$

and using standard methods [BGS75] one can construct an A such that

$$\mathrm{IP}^{\mathrm{A}} \neq \mathrm{PSPACE}^{\mathrm{A}}.$$

This shows that the IP =? PSPACE problem has contradictory relativizations and that the IP = PSPACE proof does not relativize. Similarly, we can see that the MIP =? NEXP problem has contradictory relativizations. In the following we show that these theorems also supply counterexamples to the Random Oracle Hypothesis.

3.1 ...it isn't right ...

Theorem 1 $\mathrm{Prob}_A[\ \mathrm{PSPACE}^{A} \not\subseteq \mathrm{IP}^{\mathrm{A}}\ ] = 1.$

Proof: For each oracle A, define the language:

$$\mathcal{L}(A) = \{1^n \ :\ |A^{=n}| \text{ is odd}\}.$$

Clearly, $\mathcal{L}(A) \in \mathrm{PSPACE}^{A}$ for all A. We will show that for any verifier V, the set

$$C = \{A : V^{A} \text{ is an IP verifier for } \mathcal{L}(A)\}$$

has measure zero. (C stands for "correct".) Let $p(n)$ be an upper bound on the running time on V. Fix n to be large enough so that $p(n) < \frac{1}{300}2^n$ and let $\mathcal{P}$ be the power set of $\Sigma^{\leq p(n)}$. Since $\mu(C) \leq |C \cap \mathcal{P}|/|\mathcal{P}|$, we can concentrate on counting the finite sets in $\mathcal{P}$.

Define the following classes of oracles:

$$\begin{aligned}
&\mathrm{ODD} = \{\mathrm{A} \in \mathcal{P} \ : \ |\mathrm{A}^{=\mathrm{n}}| \text{ is odd}\},\\
&\mathrm{EVEN} = \{\mathrm{A} \in \mathcal{P} \ : \ |\mathrm{A}^{=\mathrm{n}}| \text{ is even}\},\\
&S_0 = \mathrm{ODD} \cap \mathrm{C},\\
&S_1 = \mathrm{EVEN} \cap \overline{C}.
\end{aligned}$$

ODD is the set of oracles where V should accept 1^n; and EVEN the set where V should reject. S_0 is the set of oracles where V should accept and does; S_1 the set oracles where V should reject but does not.

The idea of the proof is simple. We show that $0.99 \cdot |S_0|$ is a lower bound on the size of S_1. This means that the number of oracles where V messes up is bounded below by roughly one-half of all the oracles in $\mathcal{P}$, because

$$\begin{aligned}
|\mathcal{P} \cap \overline{C}| &= |\mathrm{ODD} \cap \overline{C}| + |\mathrm{S}_1|\\
&\geq |\mathrm{ODD} \cap \overline{C}| + 0.99 \cdot |\mathrm{S}_0|\\
&\geq 0.99 \cdot (|\mathrm{ODD} \cap \overline{C}| + |\mathrm{ODD} \cap \mathrm{C}|)\\
&= 0.495 \cdot |\mathcal{P}|.
\end{aligned}$$

Thus, $|\mathcal{P} \cap C| < 0.51|\mathcal{P}|$ which tells us that the probability (over A) that V^A is part of an IP protocol for $\mathcal{L}(A)$ is less than 0.51. By repeating this proof for larger and larger n, we can drive the probability down to zero.

So, all we need to show now is that $0.99 \cdot |S_0|$ is a lower bound on the size of S_1. For every oracle $A \in S_0$, there is some prover P, which convinces V^A to accept in two thirds of the computation paths. From P and A we can construct many oracles A' such that $A' \in$ EVEN but P still convinces $V^{A'}$ to accept in at least one third of the computation paths. (This implies $A' \in S_1$.) To construct such an A' we simply add or remove a single string z from A. If P convinces $V^{A'}$ to accept 1^n in less than one third of the computation paths, then z must have appeared in more than one third of the computation paths of the $V^A(1^n)$ computation. However, the number of strings queried in at least one third of the computation paths is bounded by $3p(n)$. So, we can construct $2^n - 3p(n) \geq 0.99 \cdot 2^n$ oracles in S_1.

Formally, define the function η : ODD → EVEN by

$$\eta(A, z) = \begin{cases} A - \{z\} & \text{if } z \in A \\ A \cup \{z\} & \text{if } z \notin A. \end{cases}$$

For each $A \in S_0$ define the set R_A to be

$$R_A = \{z \ : \ |z| = n, \ \eta(A, z) \in S_1\}.$$

We claim that $|R_A| \geq 0.99 \cdot 2^n$. To see this, let $r = 2^n - |R_A|$. As mentioned before, if $|z| = n$ and $z \notin R_A$, then z must be queried in at least one third of the computation paths of $V^{\eta(A,z)}$. So, the total number of queries to strings not in R_A is at least $\frac{1}{3}r2^{p(n)}$.

However, the total number of queries in the entire computation tree is at most $p(n)2^{p(n)}$. Thus, $r \leq 3p(n)$. Since we assumed that $p(n) \leq \frac{1}{300}2^n$, $|R_A| \geq 0.99 \cdot 2^n$.

Now, define

$$D = \bigcup_{A \in S_0} \{(A, z) \mid z \in R_A\}.$$

Clearly, $|D| \geq 0.99 \cdot 2^n |S_0|$. For each $(A, z) \in D$, $\eta(A, z) \in S_1$. However, η is not a one to one function, so $|D|$ is not a lower bound on $|S_1|$. In any case, notice that $|\eta^{-1}(A')| \leq 2^n$, because $\eta(A, z) = A'$ implies that A and A' differ at only one string of length n and there are only 2^n strings on length n. Now, let's consider the total number of distinct oracles in S_1 constructed.

$$D' = \{A' \ : \ \exists (A, z) \in D, \ \eta(A, z) = A'\}.$$

An easy lower bound on $|D'|$ is $|D|/2^n$. So, we get $|D'| \geq 0.99 \cdot |S_0|$. Since $D' \subseteq S_1$, $|S_1| \geq 0.99 \cdot |S_0|$. □

Using standard techniques [BGS75,BG81,FS88], the proof of Theorem 1 can be modified to yield the following theorems. For the sake of brevity, we omit the proofs.

Theorem 2 $\text{Prob}_A[\ \text{co-NP}^A \not\subseteq \text{IP}^\text{A}\ ] = 1$.

Theorem 3 $\text{Prob}_A[\ \text{NEXP}^A \not\subseteq \text{MIP}^A\ ] = 1$.

3.2 ...it isn't even wrong.

The IP = PSPACE and MIP = NEXP results provided natural examples against the Random Oracle Hypothesis. To give a more complete understanding of the behavior of these classes with random oracles, we define a less restrictive acceptance criterion for interactive protocols and denote the class of such languages by IPP. (See also [Pap83]). We show that

$$\forall A, \ \text{IPP}^\text{A} = \text{PSPACE}^A.$$

Using the theorem in the previous section, we can provide both an example and a counterexample to the Random Oracle Hypothesis.

$$\text{Prob}_A[\ \text{IP}^\text{A} \neq \text{PSPACE}^A\ ] = 1 \quad \text{and} \quad \text{Prob}_A[\ \text{IPP}^\text{A} = \text{PSPACE}^A\ ] = 1.$$

This severely damages the already battered hypothesis because it shows that the Random Oracle Hypothesis is sensitive to small changes in the definition of complexity classes. Thus, it cannot be used to predict what happens in the real world.

Definition IPP: Let V be a probabilistic polynomial time machine and let P be an arbitrary TM. P and V share the same input tape and they communicate via a communication tape. V forms an unbounded interactive protocol for a language L if

1. $x \in L \Longrightarrow \text{Prob}[\ P\text{-}V \text{ on } x \text{ accept}\ ] > \frac{1}{2}$.
2. $x \notin L \Longrightarrow \forall P^*, \text{Prob}[\ P^*\text{-}V \text{ on } x \text{ accept}\ ] < \frac{1}{2}$.

A language L is said to be in the class IPP if it has an unbounded interactive protocol.

```
procedure CHECKCOMP($C_1, C_2, s$) ;
{ This procedure tries to detect if $M^A$ can reach configuration $C_2$ from configuration $C_1$
in $s$ steps. }
begin
    if $s = 1$ then
        { This may involve querying the oracle. }
        if $C_1 \rightarrow C_2$ in one step then
            accept
        else
            reject
    else
        Ask the prover for the middle configuration $C_3$ between $C_1$ and $C_2$.
        Toss a coin.
        if the coin toss is heads then
            CHECKCOMP($C_1, C_3, s/2$)
        else
            CHECKCOMP($C_3, C_2, s/2$)
end { procedure }
```

Figure 2: Pseudo-code for procedure CHECKCOMP.

Theorem 4 *For all oracles A,* $\mathrm{IPP}^{\mathrm{A}} = \mathrm{PSPACE}^{A}$.

Proof:
$\mathrm{IPP}^{\mathrm{A}} \subseteq \mathrm{PSPACE}^{A}$: Let L be a language in $\mathrm{IPP}^{\mathrm{A}}$. Using standard techniques [GS86, Con87], it can be shown that $\mathrm{IPP}^{\mathrm{A}}$ with private coins is the same as $\mathrm{IPP}^{\mathrm{A}}$ with *public* coins. Protocols with public coins are easy to simulate because the responses from the prover can be guessed. In fact, a PSPACE^{A} machine can traverse the entire probabilistic computation tree and compute the acceptance probability. Thus, a PSPACE^{A} machine can determine if there is a prover which makes the verifier accept in more than half of the computation paths. So, $\mathrm{IPP}^{\mathrm{A}} \subseteq \mathrm{PSPACE}^{A}$.

$\mathrm{PSPACE}^{A} \subseteq \mathrm{IPP}^{\mathrm{A}}$: This proof is similar to the proof that $\mathrm{NP} \subseteq \mathrm{PP}$ [Gil77]. Let L be a language in PSPACE^{A}. Then there is a machine M^A accepting L which runs in space $p(n)$ and halts in exactly $2^{q(n)}$ steps for some polynomials p and q. We claim that the following verifier V forms an unbounded interactive protocol for L. Given input x where $|x| = n$ do the following:

1. Toss $q(n) + 1$ coins. If all of them are "heads", then goto step 3.

2. Toss another coin. If "heads" then reject; otherwise, goto step 3.

3. Let I and F be the unique initial and final configurations of $M^A(x)$.
   Run CHECKCOMP($I, F, 2^{q(n)}$).

In order to prove the correctness of the protocol, we need the following lemma.

Lemma: Let WRONG(C_1, C_2, s) be the proposition that configuration C_2 does not follow from configuration C_1 in exactly s steps. Then,

$$\begin{aligned}&\mathrm{WRONG}(C_1, C_2, s) \Longrightarrow \\ &\quad \forall u,\ 0 \leq u \leq s,\ C_3,\ \mathrm{WRONG}(C_1, C_3, u) \bigvee \mathrm{WRONG}(C_3, C_2, s-u).\end{aligned}$$

Now, if $x \in L$, then with the prover that always tells the truth, the probability of acceptance is the same as the probability that the verifier reaches step 3, which is greater than 1/2. On the other hand, if $x \notin L$, then Prob[V rejects x | V reaches step 3] $\geq 2^{-q(n)}$. (This follows from the lemma.) In this case,

$$\begin{aligned}\text{Prob [} V \text{ rejects } x \text{]} &= \text{Prob[} V \text{ rejects } x \text{ at step 2]} \\ &\quad + \text{Prob[} V \text{ reaches step 3]} \\ &\quad \cdot \text{Prob[} V \text{ rejects } x \mid V \text{ reaches step 3]} \\ &\geq (\tfrac{1}{2} - 2^{-q(n)-2}) + (\tfrac{1}{2} + 2^{-q(n)-2}) \cdot 2^{-q(n)} \\ &> \tfrac{1}{2}.\end{aligned}$$

Therefore, Prob[V accepts x] = 1 - Prob[V rejects x] $< \frac{1}{2}$. □

4 Conclusion

We have shown that probability 1 results do not reliably predict the base case behavior of complexity classes. On the other hand, the meaning of probability 1 results for random oracles needs to be clarified and remains an interesting problem. It would be very interesting to know if there are identifiable problem classes for which the probability 1 results do point in the right direction.

In addition, we would like to note that the IP = PSPACE and MIP = NEXP results demonstrated *equality* in the base case. In many other problems with contradictory relativizations, we expect the unrelativized complexity classes to be different (e.g., we expect that P $\neq$ NP $\neq$ PSPACE, etc). The next big challenge for complexity theorists is to resolve one of these problems and *separate*—if not P and NP—any two classes with contradictory relativizations.

References

[BFL90] L. Babai, L. Fortnow, and C. Lund. Non-deterministic exponential time has two-prover interactive protocols. Technical Report 90-03, Department of Computer Science, University of Chicago, March 1990.

[BG81] C. Bennett and J. Gill. Relative to a random oracle A, $P^A \neq NP^A \neq$ co-NP^A with probability 1. *SIAM Journal on Computing*, 10(1):96–113, February 1981.

[BGS75] T. Baker, J. Gill, and R. Solovay. Relativizations of the P =? NP question. *SIAM Journal on Computing*, 4(4):431–442, December 1975.

[BH77] L. Berman and J. Hartmanis. On isomorphism and density of NP and other complete sets. *SIAM Journal on Computing*, 6:305–322, June 1977.

[Cai86] J. Cai. With probability one, a random oracle separates PSPACE from the polynomial-time hierarchy. In *ACM Symposium on Theory of Computing*, pages 21–29, 1986.

[Cai87] J. Cai. Probability one separation of the Boolean Hierarchy. In *4th Annual Symposium on Theoretical Aspects of Computer Science*, Springer-Verlag *Lecture Notes in Computer Science # 247*, pages 148–158, 1987.

[Con87] A. Condon. Computational models of games. Technical Report 87-04-04, Department of Computer Science, University of Washington, 1987.

[FS88] L. Fortnow and M. Sipser. Are there interactive protocols for co-NP languages? *Information Processing Letters*, 28(5):249–251, August 1988.

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

[GMW86] O. Goldreich, S. Micali, and A. Widgerson. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design. In *Proceedings IEEE Symposium on Foundations of Computer Science*, pages 174–187, 1986.

[Gol89] S. Goldwasser. Interactive proof systems. In *Computational Complexity Theory*, Proceedings of Symposia in Applied Mathematics, Volume 38, pages 108–128. American Mathematical Society, 1989.

[GS86] S. Goldwasser and M. Sipser. Private coins versus public coins in interactive proof systems. In *ACM Symposium on Theory of Computing*, pages 59–68, 1986.

[Har85] J. Hartmanis. Solvable problems with conflicting relativizations. *Bulletin of the European Association for Theoretical Computer Science*, 27:40–49, Oct 1985.

[Hop84] J. E. Hopcroft. Turing machines. *Scientific American*, pages 86–98, May 1984.

[Kad88] J. Kadin. The polynomial time hierarchy collapses if the Boolean hierarchy collapses. *SIAM Journal on Computing*, 17(6):1263–1282, December 1988.

[KL80] R. Karp and R. Lipton. Some connections between nonuniform and uniform complexity classes. In *ACM Symposium on Theory of Computing*, pages 302–309, 1980.

[KMR89] S. A. Kurtz, S. R. Mahaney, and J. S. Royer. The isomorphism conjecture fails relative to a random oracle. In *ACM Symposium on Theory of Computing*, pages 157–166, 1989.

[Ko88] K. Ko. Relativized polynomial time hierarchies having exactly k levels. In *ACM Symposium on Theory of Computing*, pages 245–253, 1988.

[Kur82] S. A. Kurtz. On the random oracle hypothesis. In *ACM Symposium on Theory of Computing*, pages 224–230, 1982.

[LFKN89] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. The polynomial-time hierarchy has interactive proofs. Unpublished manuscript, 1989.

[Mah82] S. Mahaney. Sparse complete sets for NP: Solution of a conjecture of Berman and Hartmanis. *Journal of Computer and System Sciences*, 25(2):130–143, 1982.

[Pap83] C. H. Papadimitriou. Games against nature. In *Proceedings IEEE Symposium on Foundations of Computer Science*, pages 446–450, 1983.

[Sha89] A. Shamir. IP = PSPACE. Unpublished manuscript, 1989.

[Yao85] A. C. Yao. Separating the polynomial-time hierarchy by oracles. In *Proceedings IEEE Symposium on Foundations of Computer Science*, pages 1–10, 1985.