An Example of a Theorem that has Contradictory Relativizations and a Diagonalization Proof

Richard Chang*

TR 89-1059
November 1989

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

*Supported in part by NSF research grant CCR-88-23053.

An Example of a Theorem that has Contradictory Relativizations and a Diagonalization Proof

Richard Chang *
Computer Science Department
Cornell University, Ithaca, NY 14853

November 27, 1989

Abstract. We construct a computable space bound $S(n)$, with $n^2 < S(n) < n^3$ and show by diagonalization that

$$\mathrm{DSPACE}[S(n)] = \mathrm{DSPACE}[S(n) \log n].$$

Moreover, we show that there exists an oracle A such that

$$\mathrm{DSPACE}^A[S(n)] \neq \mathrm{DSPACE}^A[S(n) \log n].$$

This is a counterexample to the belief that if a theorem has contradictory relativizations, then it cannot be proved using standard techniques like diagonalization [7].

1 Introduction

The study of contradictory relativizations began when Baker, Gill and Solovay [1] showed that there exist oracles A and B such that $\mathrm{P}^A \neq \mathrm{NP}^A$ and $\mathrm{P}^B = \mathrm{NP}^B$. Since contradictory relativizations had just been invented, Baker, Gill and Solovay had to explain what they thought the contradictory relativizations meant.

> We feel that this is further evidence of the difficulty of the $\mathcal{P} =?\mathcal{NP}$ question. ...It seems unlikely that ordinary diagonalization methods are adequate for producing an example of a language in $\mathcal{NP}$ but not in $\mathcal{P}$; such diagonalizations, we would expect, would apply equally well to the relativized classes. [1]

After Baker, Gill and Solovay's seminal paper, many more questions about the equality of complexity classes were shown to have contradictory relativizations. Moreover, all of these questions in their unrelativized form seemed immune from solution by standard techniques like diagonalization. Thus, the following "meta-theorem" was formulated:

> No problem that has been relativized in two conflicting ways has yet been solved, and this fact is generally taken as evidence that solutions to such problems are beyond the current state of mathematics. [7]

There have even been suggestions that the converse of this "meta-theorem" might hold.

*Supported in part by NSF research grant CCR-88-23053.

> These results show very clearly that if $LOG = NLOG$ then we cannot obtain contradictory relativizations ...and suggests that the $LOG \stackrel{?}{=} NLOG$ problem may be solvable by known techniques. [5]

However, not all of the evidence is in support of the "meta-theorem". Hartmanis [4] showed that the "meta-theorem" fails in the case of "twice relativized problems". In fact, even Baker, Gill and Solovay's interpretation of the contradictory relativizations for P and NP is in dispute:

> It has been argued that P and NP, although they may be different, are nevertheless too close together to admit separation by diagonalization. The justification for this standpoint is that known diagonalization arguments relativize. ...Nevertheless, the following result shows that if P $\neq$ NP is provable at all, then it is provable by diagonalization ...[8]

All of this paints a rather disturbing picture. The study of contradictory relativization is a central theme in structural complexity theory, but we do not have a clear understanding of its *meaning*. In this paper, we make matters worse. We provide an example of a theorem which can be proved by an easy diagonalization argument but also has contradictory relativizations.

2 One Example

Theorem 1 There exists a computable space bound $S(n)$ and an oracle A such that, $n^2 \leq S(n) < n^3$,

$$\mathrm{DSPACE}[S(n)] = \mathrm{DSPACE}[S(n)\log n],$$

but

$$\mathrm{DSPACE}^A[S(n)] \neq \mathrm{DSPACE}^A[S(n)\log n].$$

Proof:

We construct $S(n)$ so that no Turing machine uses space between $S(n)$ and $S(n)\log n$. At step n of the construction, we compute $S(n)$ by diagonalizing over a slowly growing list of Turing machines which use at most n^3 space. We make this list of machines grow so slowly that it contains only the first $r(n) - 1$ machines, where

$$r(n) = \frac{\log n}{\log\log n}.$$

Consider the following function:

$$f(i, n) = n^2 \log^i n.$$

Note that $f(i+1, n) = f(i, n)\log n$ and that $f(r(n), n) \leq n^3$. For each i, $0 \leq i < r(n)$, $f(i, n)$ is possible value for $S(n)$. The only requirement we need to satisfy is that no machine on the list uses space between $f(i, n)$ and $f(i, n)\log n$. Since there are only $r(n) - 1$ machines, by the Pigeonhole Principle, there must be some i_0, $0 \leq i_0 < r(n)$ such that no machine's space bound falls between $f(i_0, n)$ and $f(i_0 + 1, n) = f(i_0, n)\log n$. Let $S(n) = f(i_0, n)$. To find this particular i_0, simply compute and record the maximum amount of space that each machine uses for any input of length n. (This is computable since the machines are space bounded.) Then, find the first i_0 that satisfies the requirement.

Thus, every machine which uses less than n^3 space either uses less than $S(n)$ space cofinitely often or uses greater than $S(n)\log n$ space infinitely often. So, $\mathrm{DSPACE}[S(n)] = \mathrm{DSPACE}[S(n)\log n]$.

This construction does not violate the Space Hierarchy Theorem [6,9] because while $S(n)$ is easily computable in PSPACE, it is not fully space constructible.[1] In fact, this theorem is an application of the Gap Theorem [3,10].

Now, consider the oracle

$$A = \{k\#n \mid k \leq S(n)\}.$$

$S(n)$ becomes space constructible with A as an oracle. So, by the Space Hierarchy Theorem,

$$\mathrm{DSPACE}^A[S(n)] \neq \mathrm{DSPACE}^A[S(n)\log n]. \qquad \Box$$

Two observations are worth noting. First, the construction above works just as well for nondeterministic space, deterministic time and nondeterministic time. For the nondeterministic cases, the oracle A will also provide a language which diagonalizes over $\mathrm{NSPACE}^A[S(n)]$ and $\mathrm{NTIME}^A[S(n)]$. Second, $S(n)$ constructed above may not be monotonic. However, the same construction restricting $S(n)$ to be between 2^{n^2} and $2^{(n+1)^2}$ will give a monotonic space bound with the same properties.

3 Another Example

The primary objection to the previous theorem is that $S(n)$ is not space constructible. Perhaps the "meta-theorem" can be reformulated for constructible classes. The following theorem addresses this objection. It shows that oracle relativizations can be very sensitive and can pull apart classes that we would normally consider identical.

Theorem 2 There exists an oracle A such that

$$\mathrm{DSPACE}^A[n] \neq \mathrm{DSPACE}^A[2n].$$

Proof:

Recall that the Tape Compression Theorem, which shows that $\mathrm{DSPACE}[n] = \mathrm{DSPACE}[2n]$, is proved by direct simulation—a $\mathrm{DSPACE}[n]$ machine can simulate a $\mathrm{DSPACE}[2n]$ machine by using a larger alphabet. However, the expanded tape alphabet will not help in oracle queries because the oracle language uses a fixed alphabet.[2] Thus, a $\mathrm{DSPACE}^A[n]$ machine can only query strings of length n, while a $\mathrm{DSPACE}^A[2n]$ machine can query strings of length $2n$. This difference is enough to run a Baker-Gill-Solovay style diagonalization.

For any oracle A, define

$$L(A) = \{0^n \mid 0^n 1^n \in A\}.$$

Of course, $L(A) \in \mathrm{DSPACE}^A[2n]$ for any A. To construct A so $L(A) \notin \mathrm{DSPACE}^A[n]$, simply spite one $\mathrm{DSPACE}^A[n]$ machine at each length n by putting $0^n 1^n$ in A iff the machine rejects. $\Box$

There are two interpretations of Theorem 2. In one interpretation, we can deny that the Tape Compression Theorem has any real value. After all, the theorem does not hold for computers in the real world, because we cannot arbitrarily enlarge the alphabet of physical computers. So, it should not be surprising to find an oracle world where the Tape Compression Theorem does not

[1] A space bound $S(n)$ is fully space constructible if there exists a Turing machine that uses exactly $S(n)$ tape cells on input 1^n.

[2] In the standard oracle query model, the oracle tape counts as part of the work tape and is limited by the same space bound.

hold. On the other hand, we can conclude that oracle separations are just too powerful. If oracles can pull apart classes that we know are equal, then perhaps the standard model for relativizing complexity classes is too sensitive to the oracle query mechanisms and should be revised to reflect the computational powers of the base machines. (See [2] for a discussion on positive relativizations.)

4 Conclusion

The theorems presented above are not significant by themselves. Instead, they serve as counterexamples to the folklore that if a theorem has contradictory relativizations then we cannot prove it using current techniques. Without some counterexamples this folklore may very well become a self-fulfilling prophecy, because it encourages researchers to either prove a theorem (with current techniques, of course) or to show that the statement has contradictory relativizations—never both.

References

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

[2] R. V. Book. Towards a theory of relativizations: Positive relativizations. In *4th Symposium on Theoretical Aspects of Computer Science*, pages 1–21, 1987. Springer Verlag *Lecture Notes in Computer Science #247*.

[3] A. Borodin. Computational complexity and the existence of complexity gaps. *Journal of the ACM*, 19(1):158–174, 1972.

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

[5] J. Hartmanis, R. Chang, J. Kadin, and S. Mitchell. Some observations about relativization of space bounded computations. *Bulletin of the European Association for Theoretical Compter Science*, 35:82–92, June 1988.

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

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

[8] D. Kozen. Indexings of subrecursive classes. *Theoretical Computer Science*, 11:277–301, 1980.

[9] R. E. Stearns, J. Hartmanis, and P. M. Lewis II. Hierarchies of memory limited computations. In *Proceedings of the Sixth Annual Symposium on Switching Circuit Theory and Logical Design*, pages 179–190, 1965.

[10] B. A. Trakhtenbrot. *Complexity of Algorithms and Computation* [Russian]. Novosibrisk, 1967.