Yale University
Department of Computer Science

PP is Closed Under Intersection

Richard Beigel
Yale University

Nick Reingold
Yale University

Daniel Spielman
Yale College

YALEU/DCS/TR-803
June 1990

PP is Closed Under Intersection*

preliminary report

Richard Beigel† Yale University

Nick Reingold† Yale University

Daniel Spielman‡ Yale College

Abstract

In his seminal paper on probabilistic Turing machines, Gill [Gil77] asked whether the class PP is closed under intersection and union. We give a positive answer to this question. In fact, PP is closed under polynomial-time multilinear reductions. In circuits, this allows us to combine several threshold gates into a single threshold gate, while increasing depth by only a constant. Consequences in complexity theory include definite collapse and plausible separation of certain query hierarchies over PP. Consequences in the study of circuits include the simulation of circuits with a small number of threshold gates by circuits having only a single threshold gate at the root (perceptrons), and a lower bound on the number of threshold gates needed in order to compute the parity function.

1. Introduction

The class PP was defined by John Gill [Gil77] and Janos Simon [Sim75]. PP is the class of languages accepted by a polynomial-time bounded nondeterministic Turing machine that accepts when more than half of its paths are accepting and rejects when more than half of its paths are rejecting (ties can be eliminated by standard techniques). Gill noted that PP is closed under complement, but stated that it was not known if PP is closed under intersection and union. Though other closure properties of PP were later proved [Rus85, BHW89] and numerous researchers studied the class [Wag86, PY84, BDG88, Tor88, KSTT89, Tod89, Tod88, GNW90, ABFR90], Gill's question remained open, and it was widely conjectured that PP was not closed under intersection or union.

We prove that PP is in fact closed under intersection and union and even under polynomial-time conjunctive and disjunctive reductions. Consequently, PP is closed

*The authors may be reached by writing to Department of Computer Science, P.O. Box 2158, Yale Station, New Haven, CT 06520-2158, or by sending electronic mail to lastname-firstname@cs.yale.edu.

†Supported in part by NSF grants CCR-8808949 and CCR-8958528.

‡Supported in part by NSF grant CCR-8958528 under an REU supplement.

under polynomial-time truth-table reductions in which the truth table predicate is computed by a bounded-depth Boolean formula, and hence under polynomial-time Turing reductions that make $O(\log n)$ queries. That is, $\mathrm{P}^{\mathrm{PP}}_{O(\log n)\text{-T}} = \mathrm{PP}$. Relative to oracles, this collapse cannot be extended to a larger number of queries. For *functions* computed with a bounded number of queries the behavior is quite different: $\mathrm{PF}^{\mathrm{PP}}_{(k+1)\text{-tt}} \not\subseteq \mathrm{PF}^{X}_{k\text{-T}}$ for any oracle X unless P = PP.

Our strongest closure property is that PP is closed under polynomial-time truth-table reductions in which the truth predicate is computed by an explicitly produced multilinear polynomial (this includes all symmetric functions as a special case).

Similar results hold for circuits with a single threshold at the root. If g is computed by a depth-d circuit with a symmetric gate at the output, polylog n threshold gates at the next level, and $2^{\mathrm{polylog}\, n}$ AND, OR, and NOT gates at the remaining levels, then g is computed by a depth-$(d+2)$ circuit having a single threshold gate at the root and $2^{\mathrm{polylog}\, n}$ AND, OR, and NOT gates at the remaining levels. We also prove that no constant depth circuit with $o(\log n)$ threshold gates, $2^{n^{o(1)}}$ AND, OR, and NOT gates (in arbitrary positions), and $2^{n^{o(1)}}$ wires can compute parity. This is the first natural example of a function that is known to require more than a constant number of threshold gates.

2. Polynomials

Definition 1. For a non-deterministic Turing machine, N, let ACCEPT(N,x) denote the number of accepting paths of N on input x, and REJECT(N,x) be the number of rejecting paths of N on input x.

Definition 2. A language L is in PrTIME$(t(n))$ if there exists a $t(n)$-time bounded nondeterministic Turing machine N such that for all strings X, if $X \in L$ then ACCEPT(N,X) − REJECT$(N,X) > 0$, and if $X \notin L$ then ACCEPT(N,X) − REJECT$(N,X) < 0$.

Definition 3. PP = PrTIME$(n^{O(1)})$.

Beigel, Gill, and Hertrampf [BGH90] and Gundermann, Nasser, and Wechsung [GNW90] have used polynomials in order to prove closure properties of various counting classes defined by nondeterministic Turing machines. We begin this section by analyzing some of the techniques of the former paper and applying them as well to circuits.

A *threshold gate* outputs true if more than half of its inputs are true, false otherwise. By *perceptron* we mean a circuit with a single threshold gate at the output and with constant-depth Boolean circuits as inputs to the threshold (all gates are assumed to have unbounded fanin). (Green [Gre90] calls these PP(PH) circuits. Minsky and

Papert [MP88] and others define perceptrons more generally.) *Size* and *depth* of perceptrons are defined as for general circuits (for convenience we take size to be the number of wires). The *top fanin* of a perceptron is the fanin of the threshold gate. For convenience of exposition, we will use -1 to represent *false* and 1 to represent *true* in our circuits — a slight departure from standard practice. All logarithms in this paper are base 2.

Lemma 4. *For $1 \leq i \leq k$, let $x_{i,1}, \ldots, x_{i,f}$ be the outputs of an AC_0 circuit having size s and depth $D-1$, and let $x_i = \sum_{1 \leq j \leq f} x_{i,j}$. Assume that $f \geq 2$. Let $p(x_1, \ldots, x_k)$ be a polynomial of degree d, whose coefficients are integers and are bounded in absolute value by M. Then there exists a perceptron with inputs $x_{i,j}$ for $1 \leq i \leq k, 1 \leq j \leq f$, top fanin $Mf^{O(d)}(d+1)^k$, size $ks + Mf^{O(d)}(d+1)^k$, and depth $D+2$ that returns true if and only if $p(x_1, \ldots, x_k)$ is positive.*

Proof: Write each coefficient in p as a sum of 1's or a sum of -1's. Write each x_i as $x_{i,1} + \cdots + x_{i,f}$. Expand each monomial in p using the distributive law, to obtain a sum of monomials over the $x_{i,j}$'s, each monomial having coefficient -1 or 1. The number of monomials is at most $Mf^d\binom{d+k}{k}$. Each monomial has degree d or less. The value of each monomial is -1 or 1 and can be computed by a depth-2 Boolean formula (in CNF or in DNF) having size $(d+1)2^d$. Therefore there is a perceptron which determines whether $p(x_1, \ldots, x_k) > 0$ and which has top fanin $Mf^d\binom{d+k}{k}$, size $Mf^d\binom{d+k}{k} + (d+1)2^d f^d\binom{d+k}{k} + ks$ (note in this calculation that there are at most $f^d\binom{d+k}{k}$ distinct monomials, and their values need not be computed more than once), and depth $D+2$. ▮

Lemma 5. *Let $N_1, \ldots, N_k$ be $t(n)$-time bounded nondeterministic Turing machines. Let $p(x_1, \ldots, x_k)$ be a polynomial of degree d, whose coefficients are integers and are bounded in absolute value by M. Then there exists a probabilistic Turing machine N that runs in time $\lceil \log(M\binom{d+k}{k}) \rceil + dt(n)$ such that for all X, $\mathrm{ACCEPT}(N,X) - \mathrm{REJECT}(N,X) = p(x_1, \ldots, x_k)$, where x_i denotes $\mathrm{ACCEPT}(N_i,X) - \mathrm{REJECT}(N_i,X)$.*

Proof: Similar to the proof of Lemma 4. ▮

The *order* of a rational function is the degree of its numerator plus the degree of its denominator.

Lemma 6. *For $1 \leq i \leq k$, let $x_{i,1}, \ldots, x_{i,f}$ be the outputs of an AC_0 circuit having size s and depth $D-1$, and let $x_i = \sum_{1 \leq j \leq f} x_{i,j}$. Assume that $f \geq 2$. Let $r(x_1, \ldots, x_k)$ be a rational function of order d whose coefficients are integers bounded in absolute value by M. Then there exists a perceptron with inputs $x_{i,j}$ for $1 \leq i \leq k, 1 \leq j \leq f$, top fanin $M^2 f^{O(d)}(d+1)^k$, size $ks + M^2 f^{O(d)}(d+1)^k$, and depth $D+2$ such that when $r(x_1, \ldots, x_k)$ is defined the perceptron returns true if and only if $r(x_1, \ldots, x_k)$ is positive.*

Proof: Let $r(x_1, \ldots, x_k) = p(x_1, \ldots, x_k)/q(x_1, \ldots, x_k)$, and let $s(x_1, \ldots, x_k)$ be the polynomial $p(x_1, \ldots, x_k) \cdot q(x_1, \ldots, x_k)$. Then $r(x_1, \ldots, x_k)$ and $s(x_1, \ldots, x_k)$ have the same sign whenever the former is defined. The degree of s is equal to the order of r. The maximum coefficient in s is bounded in magnitude by $M^2(d+1)^{2k}$. The result follows by applying Lemma 4. ∎

Lemma 7. *Let $N_1, \ldots, N_k$ be $t(n)$-time bounded nondeterministic Turing machines. Let $r(x_1, \ldots, x_k)$ be a rational function of order d, whose coefficients are integers and are bounded in absolute value by M. Then there exists a probabilistic Turing machine N that runs in time $\lceil \log((d+1)^{2k+2}M^2) \rceil + dt(n)$ such that* $\text{ACCEPT}(N, X) - \text{REJECT}(N, X)$ *and $r(x_1, \ldots, x_k)$ have the same sign for all X where the latter is defined, where x_i denotes* $\text{ACCEPT}(N_i, X) - \text{REJECT}(N_i, X)$.

Proof: Similar to the proof of Lemma 6. ∎

The polynomial A_n defined below is the key to proving that PP is closed under intersection. When restricted to lattice points (x, y) such that $1 \le |x|, |y| \le 2^n$, the polynomial $A_n(x, y)$ is positive if and only if both x and y are positive.

$$
\begin{aligned}
P_n(x) &= (x-1)\prod_{i=1}^{n}(x-2^i)^2 \\
Q_n(x) &= -\frac{1}{2}(P_n(x) + P_n(-x)) \\
R_n(x) &= \frac{P_n(x)}{Q_n(x)} \\
&= \frac{-2P_n(x)}{P_n(x) + P_n(-x)} \\
A_n(x, y) &= 2P_n(x)Q_n(y) + 2P_n(y)Q_n(x) + Q_n(x)Q_n(y) \\
&= Q_n(x)Q_n(y)(2R_n(x) + 2R_n(y) + 1)
\end{aligned}
$$

Lemma 8. *For all n,*

i. *For all x, $P_n(x) + P_n(-x) < 0$.*

ii. *For all x, $Q_n(x) > 0$.*

iii. *If $x \le -1$ then $R_n(x) \le -2$.*

iv. *If x is an integer such that $1 \le x \le 2^n$ then $0 \le R_n(x) < 2/3$.*

Proof:

i. Since $P_n(x) + P_n(-x)$ is an even function, we may assume, without loss of generality, that $x \geq 0$. If $x < 1$ then $P_n(x)$ and $P_n(-x)$ are both negative, so the conclusion follows. If $x \geq 1$ then $P_n(x) \geq 0$ and $P_n(-x) < 0$. Therefore, it suffices to show that $|P_n(x)| < |P_n(-x)|$. Since $|x - i| < |-x - i|$ for all $i > 0$, each factor of $P_n(x)$ is greater than the corresponding factor of $P_n(-x)$, so the conclusion follows.

ii. Immediate from (i).

iii. For $x \leq -1$, $P_n(-x) \geq 0$. Therefore $P_n(x) \leq P_n(x) + P_n(-x) < 0$ by (i). Therefore $-2P_n(x)/(P_n(x) + P_n(-x)) \leq -2P_n(x)/P_n(x) = -2$.

iv. For $x \geq 1$, $P_n(x) \geq 0$. By (i), $P_n(x)+P_n(-x) < 0$. Therefore $-2P_n(x)/(P_n(x)+P_n(-x)) \geq 0$. If $x = 1$ then $R_n(x) = 0$, so the second inequality follows. Henceforth assume $x \geq 2$. We will show that $|P_n(-x)| > 4|P_n(x)|$. Since $2 \leq x \leq 2^n$ there exists k such that $2^k \leq x < 2^{k+1}$ and $1 \leq k \leq n$. For all i we have $|-x-2^i| > |x-2^i|$. We also have $|-x-2^k| \geq 2 \cdot 2^k > 2|x-2^k|$. Therefore $|P_n(-x)| > 4|P_n(x)|$, as claimed. Therefore, $-2P_n(x)/(P_n(x) + P_n(-x)) \leq -2P_n(x)/(P_n(x) - 4P_n(x)) = 2/3$, completing the proof.

∎

Lemma 9. *Assume that x and y are integers such that $1 \leq |x|, |y| \leq 2^n$. For every n,*

i. The degree of $A_n(x, y)$ is $O(n)$.

ii. Each coefficient of $A_n(x, y)$ has absolute value $O\left(2^{n^2}\right)$.

iii. If $x > 0$ and $y > 0$ then $A_n(x, y) > 0$.

iv. If $x < 0$ or $y < 0$ then $A_n(x, y) < 0$.

Proof: The assertions about the degree and size of the coefficients are easily verified.

Consider the sign of $2R_n(x) + 2R_n(y) + 1$. From Lemma 8(iii,iv) it follows that $2R_n(x) + 2R_n(y) + 1$ is positive if x and y are positive, and negative if x or y is negative. Since $Q_n(x)$ and $Q_n(y)$ are positive it follows that $A_n(x, y)$ is positive if x and y are positive, and negative if x or y is negative. ∎

For non-zero integral values of x and y within a $2^{n+1} \times 2^{n+1}$ square, $A_n(x, y)$ is positive if and only if x and y are both positive. We can construct polynomials (still with degree $O(n)$) that take on the correct sign in somewhat larger squares, though the size of the coefficients increases in that case. Minsky and Papert [MP88] have shown that no single polynomial in two variables can take on the correct sign for *all* non-zero integral values of both variables. Careful analysis of their proof shows that a polynomial in two variables of degree d with each non-zero coefficient having absolute value between 1 and M cannot take on the correct sign in a $2^{n+1} \times 2^{n+1}$ square when $n \geq \log M + d \log d$.

Next we consider analogous polynomials in many variables. Suppose that $\lceil \log k \rceil$ is odd.

Define

$$\begin{aligned}
P_n^{(k)}(x) &= P_n(x)^{\lceil \log k \rceil} \\
Q_n^{(k)}(x) &= Q_n(x)^{\lceil \log k \rceil} \\
R_n^{(k)}(x) &= \frac{P_n^{(k)}(x)}{Q_n^{(k)}(x)} \\
A_n^{(k)}(x_1, x_2, \ldots, x_k) &= 2\sum_{i=1}^{k}\left(P_n^{(k)}(x_i)\prod_{j\neq i} Q_n^{(k)}(x_j)\right) + \prod_{j=1}^{k} Q_n^{(k)}(x_j) \\
&= \left(2R_n^{(k)}(x_1) + \cdots + 2R_n^{(k)}(x_k) + 1\right)\left(Q_n^{(k)}(x_1)\cdots Q_n^{(k)}(x_k)\right)
\end{aligned}$$

Lemma 10. *For $1 \le i \le k$, assume that x_i is an integer such that $1 \le |x_i| \le 2^n$. For each n,*

i. The degree of $A_n^{(k)}(x_1, \ldots, x_k)$ is $O(n \lceil \log k \rceil k)$.

ii. Each coefficient of $A_n^{(k)}(x_1, \ldots, x_k)$ has absolute value $O\left(2^{n^2 \lceil \log k \rceil k}\right)$.

iii. If $x_i > 0$ for all i with $1 \le i \le k$, then $A_n^{(k)}(x_1, \ldots, x_k) > 0$.

iv. If $x_i < 0$ for some i with $1 \le i \le k$, then $A_n^{(k)}(x_1, \ldots, x_k) < 0$.

Proof: The first two assertions are easily verified. By Lemma 8, $R_n^{(k)}(x) \le -k$ if $x \le -1$ and $0 \le R_n^{(k)}(x) < 1$ if $1 \le x \le 2^n$. Since $Q_n^{(k)}(x)$ is positive for all x, $A_n^{(k)}(x_1, \ldots, x_k)$ and $\left(2\sum_{i=1}^{k} R_n^{(k)}(x_i) + 1\right)$ have the same sign. The last two assertions now follow easily. ∎

3. Applications to Probabilistic Turing Machines

It has been shown that PP is closed under complement [Gil77], under symmetric difference [Rus85], and under polynomial-time parity reductions [BHW89]. In this section, we will prove additional closure properties.

Theorem 11. *The union of finitely many* PrTIME$(t(n))$ *languages is in* PrTIME$(t(n)^2)$.

Proof: This follows from Lemmas 10 and 5. ∎

Torán [Tor88] asked whether PP is closed under conjunctive reductions, and he noted that an affirmative answer would imply closure under $O(\log n)$-Turing reductions (defined below).

Theorem 12. *The class* PP *is closed under polynomial-time conjunctive reductions and disjunctive reductions.*

Proof: We obtain closure under polynomial-time conjunctive reductions by applying Lemma 5 to the polynomial $A_n^{(k)}(x_1, \ldots, x_k)$ discussed in Lemma 10. We obtain closure under polynomial-time disjunctive reductions by applying Lemma 5 to the polynomial $-A_n^{(k)}(-x_1, \ldots, -x_k)$. ∎

Definition 13. A polynomial-time *bounded-depth Boolean formula reduction* is a polynomial-time truth-table reduction in which the truth-table predicate is computed by a bounded-depth Boolean formula that is explicitly produced by the reduction before any queries are made.

Theorem 14. PP *is closed under polynomial-time bounded-depth Boolean formula reductions.*

Proof: An easy induction on the depth of the formula. ∎

Definition 15. A polynomial-time $f(n)$*-Turing reduction* is a polynomial-time Turing reduction that makes at most $f(n)$ queries.

Theorem 16. PP *is closed under polynomial-time* $O(\log n)$*-Turing reductions.*

Proof: Every polynomial-time $O(\log n)$-Turing reduction can be converted to a polynomial-time depth-2 Boolean formula reduction (write the reduction as a CNF or DNF formula over the query answers). ∎

In other words, we have shown that $\mathrm{P}^{\mathrm{PP}}_{O(\log n)\text{-T}} = \mathrm{PP}$. It was previously known that $\mathrm{P}^{\mathrm{NP}}_{O(\log n)\text{-T}} \subseteq \mathrm{PP}$ [BHW89].

Definition 17. A polynomial-time *threshold reduction* is a polynomial-time truth-table reduction in which the truth-table predicate is true if and only if at least half of its inputs are true.

Definition 18. A polynomial-time *symmetric reduction* is a truth-table reduction in which the truth-table predicate is a symmetric function, i.e., a function that depends only on the number of inputs that are true.

Newman [New64] constructed rational functions that closely approximate $|x|$ for $-1 \leq x \leq 1$. To show closure under polynomial-time threshold reductions, we will define polynomials that closely approximate $sign(x)$ for $1 \leq |x| \leq 2^n$. ($sign(x) = 1$ if $x > 0$, -1 if $x < 0$, and 0 if $x = 0$.)

Recall from Lemma 8 that $0 \leq R_n(x) \leq 2/3$ for $1 \leq x \leq 2^n$ and that $R_n(x) \leq -2$ for $x \leq -1$. Define

$$\begin{aligned} \mathcal{R}_n^{(k)}(x) &= (R_n(x))^{2\lceil \log k \rceil + 2} \\ S_n^{(k)}(x) &= -1 + \frac{2}{1 + \mathcal{R}_n^{(k)}(x)}. \end{aligned}$$

Then we have $0 \leq \mathcal{R}_n^{(k)}(x) < 1/(2k)$ for $1 \leq x \leq 2^n$ and $\mathcal{R}_n^{(k)}(x) > 2k$ for $x \leq -1$. It is easily verified that $|S_n^{(k)}(x) - sign(x)| < 1/k$ when $1 \leq |x| \leq 2^n$.

Theorem 19. PP *is closed under polynomial-time threshold reductions.*

Proof: Define

$$T_n^{(k)}(x_1, \ldots, x_k) = 2S_n^{(2k)}(x_1) + \cdots + 2S_n^{(2k)}(x_k) + 1.$$

Assume that $1 \leq |x_i| \leq 2^n$ for $1 \leq i \leq k$. Then $T_n^{(k)}(x_1, \ldots, x_k)$ is a rational function that is positive if at least half of the x_i's are positive, and negative otherwise. The order of $T_n^{(k)}$ is $O(nk \log k)$, and the absolute value of each of its coefficients is $2^{O(n^2+nk^2 \log k)}$. The result now follows from Lemma 7. ∎

This immediately implies

Corollary 20. PP *is closed under polynomial-time symmetric reductions.*

Lemma 21. *Suppose $P(x_1, \ldots, x_k)$ is a multilinear polynomial in k variables with integer coefficients bounded in absolute value by M. Then there exists a rational function, $U_n(x_1, \ldots, x_k)$, of order $k(4n-1)^{2(\lceil \log(kM) \rceil + k + 3)}$, with coefficients bounded in absolute value by $O\left(2^{2n^2(\log(kM)+k+3)}\right)$, such that for $1 \leq |x_i| \leq 2^n$,*

$$|P(sign\,(x_1), \ldots, sign\,(x_k)) - U_n(x_1, \ldots, x_k)| < \frac{1}{2}.$$

Proof: Let $U_n(x_1, \ldots, x_k)$ be $P(S_n^{(h)}(x_1), \ldots, S_n^{(h)}(x_k))$, where $h = kM2^{k+2}$. The verification of the bounds on the order and coefficients is straightforward.

$P(sign\,(x_1), \ldots, sign\,(x_k))$ and $U_n(x_1, \ldots, x_k)$ have at most 2^k monomials; the error between corresponding monomials is bounded by

$$\left|M sign\,(x_1) \cdots sign\,(x_k) - MS_n^{(h)}(x_1) \cdots S_n^{(h)}(x_k)\right| < 2kM/h = 1/2^{k+1},$$

so $|P(sign\,(x_1), \ldots, sign\,(x_k)) - U_n(x_1, \ldots, x_k)| < \frac{1}{2}$. ∎

Although seemingly stronger results may be stated, note that every polynomial over variables in $\{-1, 1\}$ is equal to a multilinear polynomial over those variables.

Definition 22. A polynomial-time *multilinear reduction* is a polynomial-time truth-table reduction in which the truth-table predicate is computed by a multilinear polynomial that is explicitly produced by the reduction before any queries are made.

We now have

Theorem 23. PP *is closed under multilinear reductions.*

Note, by Lagrange's interpolation formula, that multilinear reductions include symmetric reductions as a special case. It is not known whether PP is closed under general truth-table reductions.

4. Applications to Threshold Circuits

In this section we prove some simulation results for circuits. We also prove a lower bound on the number of threshold gates needed in a constant depth circuit that computes parity.

Theorem 24. *Consider k perceptrons having top fanin f, size s, and depth D. The AND of these k perceptrons can be computed via a perceptron having top fanin $2^{O\left(k\log k(\log f)^2\right)}$, size $2^{O\left(k\log k(\log f)^2\right)} + ks$, and depth $D+2$.*

Proof: This follows from Lemma 4 and Lemma 10 with $n = \lceil \log f \rceil$. ∎

By *threshold circuit* we mean a circuit with any number of threshold, AND, OR, and NOT gates.

Lemma 25. *Consider any threshold circuit C having size s, depth D, and only k threshold gates. There is a perceptron having top fanin $2^{O\left(2^k k^3(\log k)^2(\log s)^4\right)}$, size $2^{O\left(2^k k^3(\log k)^2(\log s)^4\right)} + 2^{k+1}(k+1)s$, and depth $D+4$ which computes the same function as C.*

Proof: Number C's threshold gates $1,\ldots,k$. Let $C(b_1,\ldots,b_k)$ be the result of replacing the ith threshold gate of C by the bit b_i for every i and then evaluating the resulting threshold-free circuit. Let $A(b_1,\ldots,b_k)$ be a circuit that verifies that the result of the ith threshold gate in C is b_i for every i (using the parameters to A as the output values for any threshold gates below gate i). Negations can be pushed to the leaves, so $A(b_1,\ldots,b_k)$ can be evaluated as the AND of k thresholds. Since each of these thresholds has fanin bounded by s, $A(b_1,\ldots,b_k)$ can be computed by a perceptron having top fanin $2^{O\left(k\log k(\log s)^2\right)}$, size $2^{O\left(k\log k(\log s)^2\right)} + ks$, and depth $D+2$. The AND of $C(b_1,\ldots,b_k)$ and $A(b_1,\ldots,b_k)$ can be computed by a perceptron having top fanin $2^{O\left(k\log k(\log s)^2\right)}$, size $2^{O\left(k\log k(\log s)^2\right)} + (k+1)s$, and depth $D+2$, since $C(b_1,\ldots,b_k)$ does not involve any thresholds. The output of C can be computed by taking the OR over all 2^k sequences $b_1,\ldots,b_k$ of the AND of $A(b_1,\ldots,b_k)$ and $C(b_1,\ldots,b_k)$. This can be converted to a perceptron with top fanin $2^{O\left(2^k k^3(\log k)^2(\log s)^4\right)}$, size $2^{O\left(2^k k^3(\log k)^2(\log s)^4\right)} + 2^k(k+1)s$, and depth $D+4$. ∎

Theorem 26. *Consider any threshold circuit C having size $2^{\mathrm{polylog}\, n}$, depth D, and $O(\log\log n)$ threshold gates. There is a perceptron having top fanin $2^{\mathrm{polylog}\, n}$, size $2^{\mathrm{polylog}\, n}$, and depth $D+4$ which computes the same function as C.*

Proof: Immediate from Lemma 25. ∎

The following is known:

Theorem 27 (F. Green [Gre90]). *For any $D > 2$ there exists a constant c such that the following is true. Consider any perceptron with top fanin m, with depth $D+1$ and with subcircuits each of size $2^{n^{1/D^2}}$. If the circuit computes parity of n variables correctly, then*

$$m \geq 2^{cn^{(D+1)/D^2}-1}.$$

Using this, we show that threshold circuits with a small number of threshold gates cannot compute parity — answering a question that arose during discussions with Russell Impagliazzo. According to Roman Smolensky (personal communication, 1990) there was previously no natural example of a predicate that provably could not be computed by constant depth threshold circuits with only a constant number of threshold gates.

Theorem 28. *Let C be a threshold circuit having size $2^{n^{o(1)}}$, depth $O(1)$, and only $o(\log n)$ threshold gates. Then C does not compute the parity function of n inputs.*

Proof: By Lemma 25, C can be simulated by a perceptron having top fanin $2^{n^{o(1)}}$, size $2^{n^{o(1)}}$, and depth $O(1)$. By F. Green's theorem, such a perceptron cannot compute parity of n inputs. ∎

A *multilinear gate* evaluates a multilinear polynomial of its inputs (represented as -1 and 1) and outputs true iff the result is positive. Note, by Lagrange's interpolation formula, that every symmetric gate is a multilinear gate. Lemmas 21 and 4 yield the following simulation result:

Theorem 29. *If g is computed by a depth-D circuit with a multilinear gate at the output,* polylog n *threshold gates at the next level,* $2^{\text{polylog}\,n}$ *AND, OR, and NOT gates at the remaining levels, and* $2^{\text{polylog}\,n}$ *wires then g is computed by a perceptron that has size* $2^{\text{polylog}\,n}$ *and depth $D+2$.*

5. Query Hierarchies over PP

The class of languages polynomial-time reducible to a set A with at most $f(n)$ adaptive queries is $\mathrm{P}^A_{f(n)\text{-T}}$. The class of languages polynomial-time reducible to a set A with at most $f(n)$ nonadaptive queries is $\mathrm{P}^A_{f(n)\text{-tt}}$. The analogous classes of functions are $\mathrm{PF}^A_{f(n)\text{-T}}$ and $\mathrm{PF}^A_{f(n)\text{-tt}}$. By varying f, we obtain the query hierarchies over A.

In the preceding section we showed that $\mathrm{P}^{\mathrm{PP}}_{O(\log n)\text{-T}} = \mathrm{PP}$. This can be viewed as collapsing a query hierarchy over PP. There is an oracle A for which $\mathrm{P}^{\mathrm{NP}^A}_{f(n)\text{-T}} \subseteq \mathrm{PP}^A$ if and only if $f(n) = O(\log n)$ [Bei90]. Thus, with that same A we have $\mathrm{P}^{\mathrm{PP}^A}_{f(n)\text{-T}} = \mathrm{PP}^A$ if and only if $f(n) = O(\log n)$. Thus, there is circumstantial evidence that our collapse is the best possible. This is the first plausible example of collapse that does not translate upwards in a query hierarchy. It is also known that $\mathrm{P}^{\mathrm{PP}^R} \neq \mathrm{PP}^R$ for almost all oracles R because $\mathrm{PARITYP}^R \not\subseteq \mathrm{PP}^R$ for almost all R [ABFR90]. By combining the techniques in those two references with the techniques of this report, we can prove relativized separations at regular intervals between $O(\log n)$ queries and $n^{O(1)}$ queries. We will provide details in the final version of this report.

The behavior of the query hierarchies over PP is quite different when it comes to functions. It is known that if A is a self-reducible set that is not in P and if there exists k such that for all $j > k$ we have $\mathrm{P}^A_{j\text{-T}} = \mathrm{P}^A_{k\text{-T}}$, then A must be p-superterse, i.e., for all i and all sets X we must have $\mathrm{PF}^A_{(i+1)\text{-tt}} \not\subseteq \mathrm{PF}^X_{i\text{-T}}$ [Bei]. The class PP has self-reducible complete sets and we have $\mathrm{P}^{\mathrm{PP}}_{j\text{-T}} = \mathrm{P}^{\mathrm{PP}}_{1\text{-T}}$, so $\mathrm{PF}^A_{(i+1)\text{-tt}}$ must not be contained in $\mathrm{PF}^X_{i\text{-T}}$ for any i and X unless P = PP. This plausible separation of the bounded query hierarchies of functions is quite a contrast to the collapse discussed in the preceding paragraph. Plausible separations at higher levels in the bounded query hierarchy of functions over PP follow from [ABG90]

6. Concluding Remarks

Paturi and Saks [PS90] have also used rational approximations in their study of threshold circuits. We are grateful to them for sharing with us a preprint of their paper, in which we discovered Newman's theorem. That theorem was the inspiration for our proofs that PP is closed under symmetric reductions and under multilinear reductions (and the corresponding circuit simulations). We wish to note that the other results in this report were obtained independently of our seeing Paturi and Saks's paper.

We have exploited a relation between thresholds and rational functions. Maass and Schnitger report a relation between thresholds and sigmoidal functions $1/(1 + exp(-x))$, though their simulation requires $\omega(1)$ gates (personal communication, 1990).

We note that all previous proofs of containment in PP hinge on the existence of certain polynomials having fixed degree, in fact 2 or less in each variable. (We will discuss this further in the final version of the report.) We speculate that the need for polynomials whose degree varies with the numerical range of the inputs is the main reason why the questions answered herein remained open until now.

Our interest in the current research topic was inspired by discussions with Fred Green of the bounded query hierarchies over PP. The idea of looking at multivariate polynomials germinated during a visit from Gerd Wechsung.

We would like to thank Mike Fischer for helping us to clarify the proof of Lemma 4; John Gill, Fred Green, Rao Kosaraju, and Gerd Wechsung for helpful discussions; Andy Yao for pointing out several references, including Paturi and Saks's paper; Johan Hastad, Russell Impagliazzo, Ramamohan Paturi, and Roman Smolensky for suggesting some open problems; Steven Rudich and Mike Fischer for helpful advice; and Bill Gasarch for very helpful comments and proofreading.

The first author would also like to thank the second and third authors and Margaret for their patience.

References

[ABFR90] Jim Aspnes, Richard Beigel, Merrick Furst, and Steven Rudich, 1990. Handwritten notes.

[ABG90] Amihood Amir, Richard Beigel, and William I. Gasarch. Some connections between bounded query classes and non-uniform complexity. In *Proceedings of the 5th Annual Conference on Structure in Complexity Theory.* IEEE Computer Society Press, July 1990. To appear.

[BDG88] J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I*, volume 11 of *EATCS Monographs on Theoretical Computer Science.* Springer-Verlag, New York, 1988.

[Bei] Richard Beigel. Bounded queries to SAT and the Boolean hierarchy. *Theoretical Computer Science.* To appear.

[Bei90] Richard Beigel. Polynomial interpolation, threshold circuits, and the polynomial hierarchy. Manuscript, January 1990.

[BGH90] Richard Beigel, John Gill, and Ulrich 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, February 1990. Vol. 415 of *LNCS.*

[BHW89] Richard Beigel, Lane A. Hemachandra, and Gerd Wechsung. On the power of probabilistic polynomial time: $\mathrm{P}^{\mathrm{NP}[\log]} \subseteq \mathrm{PP}$. In *Proceedings of the 4th Annual Conference on Structure in Complexity Theory*, pages 225–227, June 1989.

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

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

[Gre90] Fred Green. An oracle separating Parity-P from PP(PH). In *Proceedings of the 5th Annual Conference on Structure in Complexity Theory.* IEEE Computer Society Press, July 1990. To appear.

[KSTT89] Johannes Köbler, Uwe Schöning, Seinosuke Toda, and Jacobo Torán. Turing machines with few accepting computations and low sets for PP. In *Proceedings of the 4th Annual Conference on Structure in Complexity Theory*, pages 208–215. IEEE Computer Society Press, June 1989.

[MP88] Marvin L. Minksy and Seymour A. Papert. *Perceptrons.* MIT Press, Cambridge, MA, 1988. Expanded Edition.

[New64] D. J. Newman. Rational approximation to $|x|$. *Michigan Mathematical Journal*, 11:11–14, 1964.

[PS90] Ramamohan Paturi and Michael E. Saks. On threshold circuits for parity. In *Proceedings of the 31st IEEE Symposium on Foundations of Computer Science*, 1990. To appear.

[PY84] C. H. Papadimitriou and M. Yannakakis. The complexity of facets (and some facets of complexity). *Journal of Computer and System Sciences*, 28:244–259, 1984.

[Rus85] D. A. Russo. *Structural Properties of Complexity Classes.* PhD thesis, University of California at Santa Barbara, March 1985.

[Sim75] Janos Simon. *On Some Central Problems In Computational Complexity.* PhD thesis, Cornell University, Ithaca, New York, 1975. Dept. of Computer Science, TR 75-224.

[Tod88] Seinosuke Toda. On probabilistic computations with a restricted access to NP oracles. Manuscript, 1988.

[Tod89] Seinosuke Toda. On the computational power of PP and P. In *Proceedings of the 30th IEEE Symposium on Foundations of Computer Science*, pages 514–519, 1989.

[Tor88] Jacobo Torán. An oracle characterization of the counting hierarchy. In *Proceedings of the 3rd Annual Conference on Structure in Complexity Theory*, pages 213–223. IEEE Computer Society Press, June 1988.

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