Electronic Colloquium on Computational Complexity, Report No. 27 (1998)

# Sparse Sets, Approximable Sets, and Parallel Queries to NP

V. Arvind
Institute of Mathematical Sciences
C. I. T. Campus
Chennai 600 113, India

Jacobo Torán
Abteilung Theoretische Informatik,
Universität Ulm,
D-89069 Ulm, Germany

**Abstract**

We relate the existence of disjunctive hard sets for NP to other well studied hypotheses in complexity theory showing that if an NP-complete set or a coNP-complete set is polynomial-time disjunctively reducible[1] to a sparse set then $\mathrm{FP}_{||}^{\mathrm{NP}} = \mathrm{FP}^{\mathrm{NP}}[\log]$. Using a similar argument we obtain also that if SAT is $O(\log n)$-approximable then $\mathrm{FP}_{||}^{\mathrm{NP}} = \mathrm{FP}^{\mathrm{NP}}[\log]$. Since $\mathrm{FP}_{||}^{\mathrm{NP}} = \mathrm{FP}^{\mathrm{NP}}[\log]$ implies that SAT is $O(\log n)$-approximable [BFT97], these two hypotheses are shown to be equivalent, thus solving an open question from [BFT97]. We show as a consequence of our first result that if an NP-complete set or a coNP-complete set is disjunctively reducible to a sparse set of polylogarithmic density then P = NP.

## 1 Introduction

The study of the existence of sparse hard sets for complexity classes has occupied complexity theorists for over two decades. The first results in this area were motivated results in this area were motivated by the Berman-Hartmanis isomorphism conjecture [BH77] and by the study of connections between uniform and nonuniform complexity classes [KL80]. The focus shifted to proving, for various reducibilities (whose strengths lie between the many-one and the Turing reducibility), that P = NP is equivalent to SAT being reducible to a sparse set via such a reducibility. It is now known (see the recent survey [CO97]) for several reducibilities that P = NP is equivalent to SAT being reducible to a sparse set via such a reducibility. A well-known result here is for the bounded truth-table reducibility [OW91]: if NP has sparse hard sets under bounded truth-table reductions then P = NP.

The possible existence of sparse Turing-hard sets for NP was considered in [KL80], while studying connections between uniform and nonuniform complexity classes. It is shown in [KL80] that if NP has sparse Turing-hard sets then the polynomial-time hierarchy (PH) collapses to $\Sigma_2^p$. In [KW95, BCGKT96] this collapse consequence is improved to $\mathrm{ZPP}^{\mathrm{NP}}$: this new collapse consequence is pretty much the best that can be proved with *relativizable techniques* since it is known that the collapse of PH (assuming NP has polynomial-size circuits) cannot be improved to $\Delta_2^p$ with relativizable techniques [IM89]. It remains a challenging open problem to prove that P = NP if there is a sparse Turing-hard set for NP. Indeed, this question remains open for stronger reducibilities .

In this paper we consider the question of existence of sparse hard sets for NP w. r. t. *disjunctive truth-table* reductions. We briefly recall some known results: it is shown in [AKM96] that if there is a sparse hard set for NP under disjunctive reductions then PH collapses to $\Delta_2^p$. More recently,

[1] All reducibilities considered in this paper are polynomial-time computable.



ISSN 1433-8092

it is shown in [CNS96] that if there are sparse hard sets for NP under the disjunctive reducibility are RP = NP. The proof technique in [CNS96] is based on powerful algebraic and randomization techniques from [CS96] that are tailored for application in the area of reductions to sparse sets. With these techniques [CS96] several long standing conjectures of Hartmanis regarding logspace and $NC^1$ reductions to sparse sets have recently been settled (see also the recent survey [CO97]).

*The results*

The contribution of this paper is to relate the question of the existence of sparse hard sets for NP under disjunctive reductions (or equivalently the question of whether the satisfiability problem SAT is disjunctively reducible to a sparse set) to other, apparently different, hypotheses in complexity theory considered in the recent work of Buhrman et. al. [BFT97]. Among these are the following four hypotheses.

(1) $\mathrm{P} = \mathrm{NP}$.

(2) $\mathrm{FP}_{||}^{\mathrm{NP}} = \mathrm{FP}^{\mathrm{NP}}[\log]$.

(3) SAT is $O(\log n)$ approximable.

(4) $(1\mathrm{SAT}, \mathrm{SAT})$ has a solution in P.

Clearly, hypothesis (1) implies the others. Furthermore, it is shown in [BFT97] that (2) implies (3). Very recently, Sivakumar [Si98], using algebraic techniques from [ALRS92], has recently shown that (3) implies (4).

It is known that RP = NP follow from (4) [VV86], and it is an outstanding open problem in structural complexity whether P = NP follows from any of hypotheses (2), (3) or (4). This is the main motivation for studying them. Cai, Naik and Sivakumar [CNS96] have shown that if SAT is disjunctively reducible to a sparse set then hypothesis (4) holds. Building on this result we show:

- If SAT or $\overline{\mathrm{SAT}}$ is disjunctively reducible to a sparse set then $\mathrm{FP}_{||}^{\mathrm{NP}} = \mathrm{FP}^{\mathrm{NP}}[\log]$.
- For any prime $k$, if $\mathrm{Mod}_k\mathrm{P}$ is disjunctively reducible to a sparse set then $(1\mathrm{SAT}, \mathrm{SAT})$ has a solution in P.

There are collapse results that follow from the hypothesis (2) that are not know to follow from (4). For example, in [JT95] it is shown that if $\mathrm{FP}_{||}^{\mathrm{NP}} = \mathrm{FP}^{\mathrm{NP}}[\log]$ then a polylogarithmic amount on nondeterminism can be simulated in polynomial time. From this follows as a corollary that if SAT or $\overline{\mathrm{SAT}}$ is disjunctively reducible to a sparse set of polylogarithmic density then P = NP. With related techniques we obtain also consequences of SAT being majority reducible to a sparse set that are discussed at the end of the paper.

With a similar argument as the one used for the results on sparse sets, applied this time to Sivakumar's proof for his main result in [Si98] we show:

- If SAT is $O(\log n)$ approximable then $\mathrm{FP}_{||}^{\mathrm{NP}} = \mathrm{FP}^{\mathrm{NP}}[\log]$.

This proves that hypotheses (2) and (3) are equivalent, answering an open question in [BFT97]. From these results we conclude that both these hypotheses are at least as weak as SAT being disjunctively reducible to a sparse set.

## 2 Preliminaries

We fix the alphabet $\Sigma = \{0,1\}$. The set $\bigcup_{0\leq i\leq n} \Sigma^i$ of all strings in $\Sigma^*$ of length up to $n$ is denoted by $\Sigma^{\leq n}$. For any set $A \subseteq \Sigma^*$, $A^{\leq n} = A \cap \Sigma^{\leq n}$, and $A^{=n} = A \cap \Sigma^n$. $\chi_A$ denotes the characteristic function of $A$. By abuse of notation, let $\chi_A(x_1, x_2, \ldots, x_m)$ denote the function that maps the list of strings $x_1, x_2, \ldots, x_m$ to the $m$-bit vector whose $i$th bit is $\chi_A(x_i)$. The length of a string $x$ is denoted by $|x|$, and the cardinality of a set $A$ is denoted by $||A||$. The density function of a set $A$ is defined as $\text{density}_A(n) = ||A^{\leq n}||$. A set $S$ is *sparse* if its density function is bounded above by a polynomial. A sparse set has *polylog density* if its density function is bounded above by $\log^k n$ for some constant $k > 0$. The complement of a language $A$ is denoted by $\overline{A}$. Let $\langle \cdot, \cdot \rangle$ denote a standard polynomial-time invertible pairing function which can be extended in a standard fashion to encode arbitrary sequences $(x_1, \ldots, x_k)$ of strings into a string $\langle x_1, \ldots, x_k \rangle$.

Unless explicitly stated all reducibilities in this paper are polynomial-time computable. Apart from the standard many-one reducibility, we consider the disjunctive truth-table reducibility: A set $A$ is *disjunctively reducible* to a set $B$, if there is a polynomial-time computable function $f$ mapping strings to sets of strings such that for all $x \in \Sigma^*$ it holds that $x \in A \iff f(x) \cap B \neq \emptyset$.

Let SAT denote the set of satisfiable boolean formulas, and let $F_{SAT}$ denote the function $\chi_{\text{SAT}}$ applied to a list of boolean formulas. We next define promise problems.

**Definition 1** [ESY84] *A promise problem is a pair of sets (Q,R). A set L is called a solution of the promise problem (Q,R) if for all $x \in Q$, $x \in L \Leftrightarrow x \in R$.*

Of particular interest to us is the promise problem $(1\text{SAT}, \text{SAT})$, where 1SAT contains precisely those boolean formulas which have at most one satisfying assignment. Observe that any solution of the promise problem $(1\text{SAT}, \text{SAT})$ has to agree with SAT in the formulas having a unique satisfying assignment as well as in the unsatisfiable formulas.

**Definition 2** [BKS95] *A function g is an f-approximator for a set A if for every $x_1, x_2, \ldots, x_m$ with $m \geq f(max_i|x_i|)$,*

$$g(x_1, x_2, \ldots, x_m) \in \Sigma^m \quad \textit{and,}$$

$$g(x_1, x_2, \ldots, x_m) \neq \chi_A((x_1, x_2, \ldots, x_m))$$

*A set A is called f-approximable if it has an f-approximator.*[2]

$\text{FP}_{||}^{\text{NP}}$ denotes the class of functions computable in polynomial time with parallel queries to an NP oracle and $\text{FP}^{\text{NP}}[\log]$ denotes the class of functions computable in polynomial time with logarithmically many adaptive queries to an NP oracle.

Other complexity-theoretic notions used in this paper can be found in textbooks like [BDG88, BDG90, Pa94].

## 3 Sparse sets and parallel queries to NP

As preparation for the first result of this paper we prove the following lemma. It is essentially based on the ideas in [CNS96] stated in a more general setting. We are interested in solving the following decoding problem which we call the *hidden polynomial* problem: Let $F_q$ denote the finite field of size $q$. Suppose there is an unknown univariate polynomial $P(x)$ of degree $n$ over $F_q$. Also,

[2] Approximability is called membership comparability in [Og95].

suppose we are given a 'noisy' table consisting of $N$ rows, with each row containing a variable-sized list of pairs $\langle u, v\rangle \in F_q \times F_q$ with the claim that $P(u) = v$. In addition, we know that there is a small set of $t$ ( such that $q \geq (n+1)t$) correct rows which completely specify the polynomial $P$ on the whole of $F_q$. The problem is to efficiently compute a small set of candidate polynomials which includes $P$.

The following lemma (based on [CNS96]) gives a precise answer to this problem.

**Lemma 3** *There is an algorithm (that runs in time polynomial in $n$, $N$, and $q$) that takes as input a table $T$ of size $N \times q^2$, as described above, and outputs a list of at most $N$ polynomials, one of which is the hidden polynomial.*

*Proof.* Notice that there are exactly $q^2$ pairs $\langle u, v\rangle$, $u, v \in F_q$ of which exactly $q$ pairs correctly define the graph of the hidden polynomial $P$. Since there is a set of $t$ correct rows in the table $T$ which completely specify the polynomial, by pigeon-hole principle there is one correct row which contains at least $q/t$ pairs. Furthermore, notice that no correct row contains *inconsistent* pairs $\langle u, v\rangle$ and $\langle u, w\rangle$ where $v \neq w$.

Call a row of the table *long* if it has at least $q/t$ pairs and does not contain any inconsistent pair. We know that there is at least one correct row which is long.

Writing the hidden polynomial $P(x)$ as $\sum_{i=0}^{n} a_i x^i$ we notice that each long row gives us a system of at least $q/t$ linear equations in the $n+1$ unknowns $a_i, 0 \leq i \leq n$. Since $q/t \geq n+1$, we can pick any $n+1$ of the equations corresponding to a given long row which will have a unique solution in the $a_i$'s since the coefficient matrix is a Vandermonde matrix which is invertible. Using Gaussian elimination we can efficiently compute this unique solution for each long row.

This yields a list of at most $N$ polynomials, one for each long row in the table $T$, and we know that the hidden polynomial (corresponding to a good long row of $T$) is in this list. ∎

We prove now the first result of the paper.

**Theorem 4** *If* SAT *is disjunctively reducible to a sparse set then* $\mathrm{FP}_{||}^{\mathrm{NP}} = \mathrm{FP}^{\mathrm{NP}}[\log]$.

*Proof.* Suppose SAT is disjunctively reducible to a sparse set. Recall that $F_{SAT}$, which computes the characteristic sequence of a list of SAT queries, is complete for $\mathrm{FP}_{||}^{\mathrm{NP}}$. It suffices to show that $F_{SAT}$ is in $\mathrm{FP}^{\mathrm{NP}}[\log]$. We will design an $\mathrm{FP}^{\mathrm{NP}}[\log]$ machine $M$ for $F_{SAT}$. On input a list of formulas $(x_1, x_2, \ldots, x_m)$, the machine $M$ first computes, with a binary search and queries to a suitable NP oracle, the cardinality $k$ of $\{x_1, x_2, \ldots, x_m\} \cap \mathrm{SAT}$.

Now consider the following set $Y = \{\langle q, u, v, k, x_1, x_2, \ldots, x_m\rangle \mid 0 \leq u, v \leq q-1$, and $\exists a \in \Sigma^m$ with $k$ 1's such that if $a_i = 1$ then $x_i \in \mathrm{SAT}$ and $\sum_{i=1}^{m} a_i u^{i-1} \equiv v \pmod q\}$.

Notice that $Y \in \mathrm{NP}$. Also, observe now that if $k$ is $||\{x_1, x_2, \ldots, x_m\} \cap \mathrm{SAT}||$ then there is a unique vector $a \in \Sigma^m$ such that if $a_i = 1$ then $x_i \in \mathrm{SAT}$. Thus, for a given triple $q, u, v$ there is at most one vector $a \in \Sigma^m$ satisfying the above property.

Actually, we are interested only in those instances $\langle q, u, v, k, x_1, x_2, \ldots, x_m\rangle$ of $Y$ where $q$ is a small prime. More precisely, consider an instance $(x_1, x_2, \ldots, x_m)$ of $F_{SAT}$ of length $n$. Corresponding to this instance, we pick $q$ to be a $c \log n$ bit prime number, where we will choose $c$ later appropriately. Let $F_q$ denote the finite field of size $q$. Notice that we can pick $q$ and construct the field $F_q$ efficiently (i.e. in time polynomial in $n$). Moreover, arithmetic in $F_q$ can also be done efficiently.

Since $Y \in \mathrm{NP}$ there is a disjunctive reduction $f$ from $Y$ to a sparse set $S$ of density $||S^{\leq n}|| \leq p(n)$ for some polynomial $p$. I.e. $f$ is an FP function that on input $x$ produces a set of strings $f(x)$ such

that $x \in Y$ iff $f(x) \cap S \neq \emptyset$. Fix an instance $(x_1, x_2, \ldots, x_m)$ of $F_{SAT}$. Let the length of this instance be $n$. Let $q$ be a $c \log n$ bit prime number. The length of $\langle q, u, v, k, x_1, x_2, \ldots, x_m \rangle$ using a standard pairing function can be bounded by $2n$ for large enough $n$ since $q, u, v$ and $k$ can be encoded in $3c \log n + \log n$ bits. Now, since the reduction $f$ from $Y$ to $S$ is polynomial-time computable there is a polynomial $r(n)$ which bounds both $||f(\langle u, v, k, x_1, x_2, \ldots, x_m \rangle)||$ and the length of each query in $f(\langle q, u, v, k, x_1, x_2, \ldots, x_m \rangle)$. Let $Q = \bigcup_{u,v \in F_q} f(\langle u, v, k, x_1, x_2, \ldots, x_m \rangle)$. Write $Q = \{q_1, q_2, \ldots, q_N\}$. Our aim is to apply Lemma 3. Build a table $T$ with $N$ rows where we put $\langle u, v \rangle$ in row $i$ if $q_i \in f(\langle q, u, v, k, x_1, x_2, \ldots, x_m \rangle)$. Note that $N \leq r(n)q^2$. We define row $i$ to be *correct* if $q_i \in S$. Notice that there are at most $||S^{\leq r(n)}|| \leq r(n)p(r(n))$ good rows in the table $T$. Now we choose the constant $c$ (which determines the size of the field $F_q$) so that $q/r(n)p(r(n)) = n^c/r(n)p(r(n)) > n$, where we know that $n \geq m$.

Let $F_{SAT}(x_1, x_2, \ldots, x_m) = a_1 a_2 \ldots a_m$. Then a hidden polynomial specified by the table $T$ is $\sum_{i=1}^{m} a_i x^{i-1}$. Applying the algorithm of Lemma 3 we can compute in time polynomial in $n$ a list $X$ of at most $N$ polynomials of degree $m-1$. Each of these polynomials gives us an $m$-bit vector of its coefficients. We discard from this list those $m$-bit vectors which have a number of 1's different from $k$. In the pruned list exactly one $m$-bit vector is $F_{SAT}(x_1, x_2, \ldots, x_m)$ and every other $m$-bit vector has a 1 at a position where the corresponding formula in $(x_1, x_2, \ldots, x_m)$ is unsatisfiable.

The $\mathrm{FP}^{\mathrm{NP}}[\log]$ machine $M$ can now find the *unique* correct $m$-bit vector in $X$ by doing a standard binary search guided by at most $\log N = O(\log n)$ queries to a suitable NP oracle. This completes the proof. ■

It is an open question whether we can derive P = NP from the assumption that SAT is disjunctively reducible to a sparse set. One direction is to consider disjunctive reductions from SAT to sets of density lower than polynomial. It is already known that if SAT is disjunctively reducible to a tally set then P = NP [Uk83, Ya83]. However, the proof technique of [Uk83, Ya83] does not work if we assume that SAT is disjunctively reducible to a set $S$ of polylog density. Reductions of SAT to sets of polylog density were considered by Buhrman and Hermo [BH95] where they show that if SAT is Turing reducible to a set of polylog density then $\mathrm{NP}(\log^k n) = \mathrm{NP}$ for all $k$ (where $\mathrm{NP}(\log^k n)$ is the class of NP languages accepted by NP machines which make at most $\log^k n$ nondeterministic moves on inputs of length $n$). We also recall here the result of Jenner and Torán [JT95] that if $\mathrm{FP}_{||}^{\mathrm{NP}} = \mathrm{FP}^{\mathrm{NP}}[\log]$ then $\mathrm{NP}(\log^k n) = \mathrm{P}$ for each $k > 0$.

Combining the above-mentioned results of [BH95, JT95] with Theorem 4 immediately yields the following corollary.

**Corollary 5** *If* SAT *is disjunctively reducible to a set of polylog density then* P = NP.

We next briefly consider consequences of SAT being conjunctively reducible to a co-sparse set. By complementation, this is equivalent to $\overline{\mathrm{SAT}}$ being disjunctively reducible to a sparse set. We show that we can apply again the technique of [CNS96] to derive $\mathrm{FP}_{||}^{\mathrm{NP}} = \mathrm{FP}^{\mathrm{NP}}[\log]$ as a consequence.

**Theorem 6** *If* $\overline{\mathrm{SAT}}$ *is disjunctively reducible to a sparse set then* $\mathrm{FP}_{||}^{\mathrm{NP}} = \mathrm{FP}^{\mathrm{NP}}[\log]$.

*Proof.* Suppose $\overline{\mathrm{SAT}}$ is disjunctively reducible to a sparse set. Again, it suffices to show that $F_{SAT}$ is in $\mathrm{FP}^{\mathrm{NP}}[\log]$. On input a list of formulas $(x_1, x_2, \ldots, x_m)$, the cardinality $k$ of $\{x_1, x_2, \ldots, x_m\} \cap \mathrm{SAT}$ can be computed with an $\mathrm{FP}^{\mathrm{NP}}[\log]$ computation.

We introduce some notation for conciseness. Let $w$ denote an assignment to all variables in $\{x_1, x_2, \ldots, x_m\}$. Let $x_i(w)$ denote the value of formula $x_i$ at assignment $w$. We define the following polynomial-time computable predicate:

$U(\langle q, u, v, k, x_1, x_2, \ldots, x_m\rangle, a, w) := (\sum_{i=1}^{m} a_i = k \wedge \bigwedge_{a_i=1} x_i(w) = 1 \wedge \sum_{i=1}^{m} a_i u^{i-1} = v) \vee (\sum_{i=1}^{m} a_i \neq k) \vee (\bigwedge_{a_i=1} x_i(w) = 0)$

Consider now the following set $Z = \{\langle q, u, v, k, x_1, x_2, \ldots, x_m\rangle \mid 0 \leq u, v \leq q-1$, and $\forall a \in \Sigma^m$ $\forall$ assignments $w$: $U(\langle q, u, v, k, x_1, x_2, \ldots, x_m\rangle, a, w)\}$

Since $U$ is a polynomial-time predicate it follows that $Z \in$ coNP.

Observe that if $k = ||\{x_1, x_2, \ldots, x_m\} \cap \mathrm{SAT}||$ then $a_i = 1$ implies $x_i \in \mathrm{SAT}$ for all $i$ iff $a \in \Sigma^m$ is the characteristic vector of $x_1, x_2, \ldots, x_m$. We are interested in instances $\langle q, u, v, k, x_1, x_2, \ldots, x_m\rangle$ of $Z$ for $q$ picked to be a small prime. If $|(x_1, x_2, \ldots, x_m)| = n$, we will pick $q$ to be a $c \log n$ bit prime number, for an appropriate $c$.

Since $Z \in$ coNP there is a disjunctive reduction $f$ from $Z$ to a sparse set $S$ of density $||S^{\leq n}|| \leq p(n)$ for some polynomial $p$. I.e. $f$ is an FP function that on input $x$ produces a set of strings $f(x)$ such that $x \in Y$ iff $f(x) \cap S \neq \emptyset$. Fix an instance $(x_1, x_2, \ldots, x_m)$ of $F_{SAT}$. Let the length of this instance be $n$. Let $q$ be a $c \log n$ bit prime number. It means that $u$ and $v$ in $\langle q, u, v, k, x_1, x_2, \ldots, x_m\rangle$ will be picked from the finite field $F_q$. As in the proof of Theorem 4 $|\langle q, u, v, k, x_1, x_2, \ldots, x_m\rangle|$ can be bounded by $2n$. There is a polynomial $r$ such that $r(n)$ bounds both $||f(\langle u, v, k, x_1, x_2, \ldots, x_m\rangle)||$ and the length of each query in $f(\langle q, u, v, k, x_1, x_2, \ldots, x_m\rangle)$.

The crucial property that we exploit is the following claim that is easy to check from the definition of $Z$.

**Claim** *If* $k = ||\{x_1, x_2, \ldots, x_m\} \cap \mathrm{SAT}||$ *then* $\langle u, v, k, x_1, x_2, \ldots, x_m\rangle \in Z$ *iff* $\sum_{i=1}^{m} a_i u^{i-1} = v$ *holds for* $a = F_{SAT}(x_1, x_2, \ldots, x_m)$.

Now, let $Q = \bigcup_{u,v \in F_q} f(\langle u, v, k, x_1, x_2, \ldots, x_m\rangle)$. Write $Q = \{q_1, q_2, \ldots, q_N\}$. In order to apply Lemma 3 we build a table $T$ with $N$ rows and put $\langle u, v\rangle$ in row $i$ if $q_i \in f(\langle q, u, v, k, x_1, x_2, \ldots, x_m\rangle)$. Note that $N \leq r(n)q^2$. We define row $i$ to be *correct* if $q_i \in S$. Notice that there are at most $||S^{\leq r(n)}|| \leq r(n)p(r(n))$ correct rows in $T$. Choose the constant $c$ (which determines the size of the field $F_q$) so that $q/r(n)p(r(n)) = n^c/r(n)p(r(n)) > n$, where we know that $n \geq m$.

As in the proof of Theorem 4 we can find a list of at most $N$ $m$-bit vectors one of which is $F_{SAT}(x_1, x_2, \ldots, x_m)$, which we can locate by doing a binary search with an $\mathrm{FP}^{\mathrm{NP}}[\log]$ computation. ■

We have the following immediate corollary.

**Corollary 7** *If* $\overline{\mathrm{SAT}}$ *is disjunctively reducible to a sparse set then there is a solution of* $(1\mathrm{SAT}, \mathrm{SAT})$ *in* P, *and consequently* NP = RP.

We next consider disjunctive reductions from $\mathrm{Mod}_k\mathrm{P}$ to sparse sets. We first briefly recall the definition of $\mathrm{Mod}_k\mathrm{P}$. For an NP machine $N$ let $\mathrm{acc}_N(x)$ denote the number of accepting computations of $N$ on input $x \in \Sigma^*$. A language $L$ is in the class $\mathrm{Mod}_k\mathrm{P}$ if there is an NP machine $N$ such that $x \in L \iff \mathrm{acc}_N(x) \not\equiv 0 (\mathrm{mod}\ k)$.

**Theorem 8** *For prime* $k$, *if a* $\mathrm{Mod}_k\mathrm{P}$*-complete set is disjunctively reducible to a sparse set then there is a solution of* $(1\mathrm{SAT}, \mathrm{SAT})$ *in* P.

*Proof.* Let $\#F$ denote the number of satisfying assignments of the formula $F$. Consider the following complete problem $\mathrm{Mod}_k\mathrm{SAT}$ for $\mathrm{Mod}_k\mathrm{P}$:

$$\mathrm{Mod}_k\mathrm{SAT} := \{F \mid \#F \equiv 1 \ (\mathrm{mod}\ k)\}$$

Let $Y$ be the language consisting of tuples $\langle q, u, v, F\rangle$ satisfying the following two properties:

- $F$ is a boolean formula and $0 \leq u, v \leq q-1$, for nonnegative integers $q, u$, and $v$.
- If $F$ is an $m$-ary boolean formula then $||\{a \in \Sigma^m \mid F(a) = 1 \text{ and } \sum_{i=1}^{m} a_i u^{i-1} = v \pmod q\}|| \equiv 1 \pmod k$

The following easy claim is the crucial observation.

**Claim:** *If $F$ is an instance of* $(1\text{SAT}, \text{SAT})$ *then* $\langle q, u, v, F\rangle \in Y$ *iff $F$ is satisfiable and* $\sum_{i=1}^{m} a_i u^{i-1} = v \pmod q$ *holds for the unique satisfying assignment $a$.*

Given an instance $F$ of $(1\text{SAT}, \text{SAT})$, we pick a $c \log n$ bit prime number $q$ (for appropriate $c$) and consider the disjunctive reduction $f$ from $Y$ to a sparse set $S$, as applied only to instances $\langle q, u, v, F\rangle$, for $u, v \in F_q$. Similar to the argument in the proof of Theorem 4, we can choose the constant $c$ appropriately large (depending on density of $S$ and the reduction $f$) so that Lemma 3 can be applied to yield a polynomially bounded set of $m$-bit vectors one of which is the unique solution of $F$, if $F$ is satisfiable. Thus, we can decide satisfiability for instances of $(1\text{SAT}, \text{SAT})$. ■

# 4 Approximability and parallel queries to NP

In this section we show that if SAT is $O(\log n)$-approximable then $\text{FP}_{||}^{\text{NP}} = \text{FP}^{\text{NP}}[\log]$. We prove this result by applying the main technical idea in the recent paper by Sivakumar [Si98] where he shows that if SAT is $O(\log n)$ approximable then the promise problem $(1\text{SAT}, \text{SAT})$ is in P.[3]

**Theorem 9** *If SAT is $O(\log n)$ approximable then* $\text{FP}_{||}^{\text{NP}} = \text{FP}^{\text{NP}}[\log]$.

*Proof.* Assume SAT is $O(\log n)$ approximable. In other words, there are a constant $c$ and an FP function $f$ such that $f(\langle x_1, x_2, \ldots, x_k\rangle)$ is a $k$-bit vector different from $F_{SAT}(x_1, x_2, \ldots, x_k)$ for any $k$-tuple of formulas $\langle x_1, x_2, \ldots, x_k\rangle$ with $k \geq c \log(max_i |x_i|)$.

As before, we'll prove the result by giving an $\text{FP}^{\text{NP}}[\log]$ machine, call it $M$, that computes $F_{SAT}$. Let $(x_1, x_2, \ldots, x_m)$ be an input instance for $F_{SAT}$. The first step of $M$ is to compute via a binary search guided by a suitable NP oracle the number $k$ of the $x_i$'s that are in SAT. As in the previous proof we will pick a suitable constant $c$ and efficiently construct the finite field $F_q$, where $q$ is a $c \log n$ bit prime number. For a binary vector $a = a_1 a_2 \ldots a_m \in \Sigma^n$ let $P_a$ denote the univariate polynomial $\sum_{i=1}^{m} a_i x^{i-1}$ over $F_q$. We define the following new language: $Z = \{\langle u, j, k, x_1, x_2, \ldots, x_m\rangle \mid \exists a \in \Sigma^m$ with $k$ 1's such that if $a_i = 1$ then $x_i \in$ SAT and $j$th bit of $P_a(u)$ is $1\}$.

Clearly, this language is in NP and is therefore $O(\log n)$ approximable by hypothesis.

The next two technical steps are exactly as in [Si98].

It is not hard to see that we can apply [Si98, Corollary, page 5] to get in polynomial time for each $u \in F_q$ a set $S_u \subseteq F_q$ such that $||S_u|| \leq q^{1/3}$ and $P_a(u) \in S_u$. Next, applying [ALRS92] (as described in [Si98]) we can efficiently reconstruct a bunch of some $N$ polynomials of degree $m-1$ that includes $P_a$ and $N$ is bounded by a polynomial in $n$, the length of $(x_1, x_2, \ldots, x_m)$.

We can recover from this list of $N$ polynomials a list of $N$ $m$-bit vectors. Afterwards we discard those vectors which contain a number of 1's different from $k$, we know that except $F_{SAT}(x_1, x_2, \ldots, x_m)$ which is in this list every other $m$-bit vector has a 1 in a position where the corresponding formula in $(x_1, x_2, \ldots, x_m)$ is unsatisfiable.

[3] Both [Si98] and [BFT97] call the promise problem UniqueSAT which can be confused with USAT. In this paper we have used Selman's notation as in [ESY84]

As described in the earlier proof we can find this vector by doing a binary search with at most $\log N$ queries to a suitable NP oracle. This completes the proof. ■

In [BFT97] it is shown that if $\mathrm{FP}_{||}^{\mathrm{NP}} = \mathrm{FP}^{\mathrm{NP}}[\log]$ then SAT is $O(\log n)$ approximable. Combined with the above theorem we have the following corollary.

**Corollary 10** *SAT is $O(\log n)$ approximable iff $\mathrm{FP}_{||}^{\mathrm{NP}} = \mathrm{FP}^{\mathrm{NP}}[\log]$.*

The following corollary is also an immediate consequence of Theorems 9, 4, and 6.

**Corollary 11**

- *If SAT or $\overline{\mathrm{SAT}}$ are disjunctively reducible to a sparse set then SAT is $O(\log n)$-approximable.*

# 5 Majority reductions to sparse sets

Finally, we have a couple of observations and an open problem regarding majority reductions to sparse sets. Majority reductions to sparse sets are interesting because they generalize both conjunctive and disjunctive reductions to sparse sets. A set $A$ is *majority reducible* to a set $B$ if there is an FP function $f$ that on input $x$ produces a set of strings $f(x)$ such that $x \in A$ iff $||f(x) \cap B|| > ||f(x)||/2$. In other words, the majority of strings in $f(x)$ are in $B$.

The following lemma is easy to prove by appropriately padding the list of queries produced by the reduction function $f$ with a suitable number of strings (we pad with copies of either a fixed string known to be in the sparse set or a fixed string known to be outside the sparse set).

**Lemma 12** *If a set $A$ is conjunctively or disjunctively reducible to a sparse set, then in fact $A$ is majority reducible to some sparse set.*

In [CNS96] *bpp*-reductions of SAT to sparse sets are considered. We recall the definition of *bpp* reducibility: A set $A$ is *bpp*-reducible to a set $B$ if there is a polynomial-time function $f$ and polynomials $p$ and $q$ such that for all $x$,

$$\begin{aligned} x \in A &\Rightarrow Prob[f(x,w) \in B] \geq 1/2 + 1/p(|x|), \text{ and} \\ x \notin A &\Rightarrow Prob[f(x,w) \notin B] \geq 1/2 + 1/p(|x|), \end{aligned}$$

where the string $w$ is chosen uniformly at random from the set $\Sigma^{q(|x|)}$. The success probability of the reduction is $1/2 + 1/p(|x|)$.

The following result is shown in [CNS96].

**Theorem 13** [CNS96] *If* SAT *is bpp-reducible to a sparse set then* NP = RP.

We next observe the following easy lemma connecting majority reductions to *bpp*-reductions in the obvious way.

**Lemma 14** *If a set $A$ is majority reducible to a set $B$ then, in fact, $A$ is bpp-reducible to $B$, with the reduction having success probability $1/2 + 1/n^{O(1)}$.*

Combining this with the above stated theorem of [CNS96] we get the following corollary.

**Corollary 15** *If* SAT *is majority reducible to a sparse set then* NP = RP.

We leave as an open question whether $\mathrm{FP}_{||}^{\mathrm{NP}} = \mathrm{FP}^{\mathrm{NP}}[\log]$ or even the weaker consequence that (1SAT, SAT) has a solution in P, can be proved from the assumption that SAT is majority reducible to a sparse set.

## References


[ALRS92] S. Ar, R. Lipton, R. Rubinfeld, and M. Sudan. Reconstructing algebraic functions from erroneous data. In *Proc. 33rd Annual IEEE Symp. on Foundations of Computer Science,* 503–512, 1992.

[AKM96] V. Arvind, J. Köbler and M. Mundhenk. Upper bounds for the complexity of sparse and tally descriptions. In *Mathematical Systems Theory,* 29:63–94, 1996.

[BDG88] J. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I.* Springer-Verlag, 1988.

[BDG90] J. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity II.* Springer-Verlag, 1990.

[BKS95] R. Beigel, M. Kummer, and F. Stephan. Approximable sets. *Information and Computation,* 120(2):304–314, 1995.

[BH77] L. Berman and J. Hartmanis. On isomorphisms and density of NP and other complete sets. *SIAM Journal on Computing,* 6(2):305–322, 1977.

[BCGKT96] N. Bshouty, R. Cleve, R. Gavaldà, S. Kannan, and C. Tamon. Oracles and queries that are sufficient for exact learning. *Journal of Computer and System Sciences*, **52**:421–433, 1996.

[BFT97] H. Buhrman, L. Fortnow, and L. Torenvliet. Six hypotheses in search of a theorem. In *Proc. 12th Annual IEEE Conference on Computational Complexity,* 2–12, IEEE Computer Society Press, 1997.

[BH95] H. Buhrman and M. Hermo. On the sparse set conjecture for sets with low density. In *Proc. 12th Annual Symp. on Theoretical Aspects of Computer Science,* Lecture Notes in Computer Science 900, 609–618, Springer Verlag 1995.

[CNS96] J. Y. Cai, A. Naik, and D. Sivakumar. On the existence of hard sparse sets under weak reductions. In *Proc. 13th Annual Symp. on Theoretical Aspects of Computer Science,* 307–318, 1996.

[CO97] J. Y. Cai and M. Ogihara. Sparse sets versus complexity classes. Chapter in *Complexity Theory Retrospective II*, L. Hemaspaandra and A. Selman editors, Springer Verlag 1997.

[CS96] J. Y. Cai and D. Sivakumar. The resolution of a Hartmanis conjecture. In *Proc. 36th Foundations of Computer Science,* 362–373, 1995.

[ESY84] S. Even, A. Selman, and Y. Yacobi. The complexity of promise problems with applications to public-key cryptography. *Information and Control*, 61:114-133, 1984.

[IM89] N. Immerman and S. Mahaney. Relativizing relativized computations. *Theoretical Computer Science*, 68:267–276, 1989.

[JT95] B. Jenner and J. Torán. Computing functions with parallel queries to NP. *Theoretical Computer Science,* 141, 175–193, 1995.

[KL80] R. M. Karp and R. J. Lipton. Some connections between nonuniform and uniform complexity classes. In *Proc. 12th ACM Symposium on Theory of Computing*, 302–309. ACM Press, 1980.

[KW95] J. Köbler and O. Watanabe. New collapse consequences of NP having small circuits. In *Proceedings of the 22nd International Colloquium on Automata, Languages, and Programming*, Lecture Notes in Computer Science #944, 196–207. Springer-Verlag, 1995.

[Ma82] 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.

[Og95] M. Ogihara. Polynomial-time membership comparable sets. *SIAM Journal of Computing,* 24(5):1168–1181, 1995.

[OW91] M. Ogihara and O. Watanabe. On polynomial time bounded truth-table reducibility of NP sets to sparse sets. *SIAM Journal on Computing* 20(3):471–483 (1991).

[Pa94] C. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

[Si98] D. Sivakumar. On membership comparable sets. To appear in the *13th IEEE Computational Complexity Conference 1998.*

[Uk83] E. Ukkonen. Two results on polynomial time truth-table reductions to sparse sets. *SIAM Journal on Computing*, 12(3):580–587, 1983.

[VV86] L. Valiant and V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47:85–93,1986.

[Ya83] C. Yap. Some consequences of non-uniform conditions on uniform classes. *Theoretical Computer Science*, 26:287–300, 1983.



ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'