Deterministic and Randomized Bounded Truth-table Reductions of P, NL, and L to Sparse Sets

Dieter van Melkebeek*
Department of Computer Science
The University of Chicago
Chicago, IL 60637

Abstract

We prove that there is no sparse hard set for P under logspace computable bounded truth-table reductions unless P = L. In case of reductions computable in NC^1, the collapse goes down to $P = NC^1$.

We parameterize this result and obtain a generic theorem allowing to vary the sparseness condition, the space bound and the number of queries of the truth-table reduction. Another instantiation yields that there is no quasipolynomially dense hard set for P under polylog-space computable truth-table reductions using polylogarithmically many queries unless P is in polylog-space.

We also apply the proof technique to NL and L. We establish that there is no sparse hard set for NL under logspace computable bounded truth-table reductions unless NL = L, and that there is no sparse hard set for L under NC^1-computable bounded truth-table reductions unless $L = NC^1$.

We show that all these results carry over to the randomized setting: If we allow two-sided error randomized reductions with confidence at least inversely polynomial, we obtain collapses to the corresponding randomized classes in the multiple access model. In addition, we prove that there is no sparse hard set for NP under two-sided error randomized polynomial-time bounded truth-table reductions with confidence at least inversely polynomial unless NP = RP.

1 Introduction

Recently, complexity theorists made much progress on the sparse hard set problem for P, i.e., the question whether there are sparse hard sets for P under various logspace computable reducibilities. Ogihara [26] showed that the existence of a sparse hard set for P under logspace computable many-one reductions implies that $P \subseteq \mathrm{DSPACE}[\log^2 n]$, and Cai and Sivakumar [8] subsequently proved that this hypothesis actually yields P = L. Cai, Naik and Sivakumar [7] next considered truth-table reductions. They argued that the existence of a sparse hard set for P under logspace computable bounded truth-table reductions implies $P = NC^2$ [6], but left the implication P = L open. We establish here as our main result that this final collapse does follow:

*Most of the research was done while the author was funded as a Research Assistant of the Belgian National Fund for Scientific Research at the Katholieke Universiteit Leuven. Partial support from the NSF through grant CCR 92-53582, from the European Union through Marie Curie Research Training Grant ERB-4001-GT-96-0783 while visiting CWI and the University of Amsterdam, and from the Fields Institute.

Theorem 1.1 (Main Theorem) *There is no sparse hard set for* P *under logspace computable bounded truth-table reductions unless* P = L.

In order to situate the sparse hard set problem for P, let us first look at the analogous problem for NP.

1.1 The Sparse Hard Set Problem for NP

Researchers have spent considerable effort investigating whether there are sparse hard sets for NP under various polynomial-time computable reducibilities. Two major issues motivate them in doing so:

- A fundamental result by Meyer [4] states that the sets polynomial-time *Turing reducible* to sparse sets are precisely those that have polynomial-size (non-uniform) circuits. The same holds for truth-table reductions, i.e., Turing reductions with non-adaptive queries. Consequently, a sparse hard set for NP under polynomial-time Turing reductions exists iff a sparse hard set for NP under polynomial-time truth-table reductions exists iff every set in NP has polynomial-size circuits.

- Berman and Hartmanis [4] observed that all known sets complete for NP under polynomial-time computable *many-one reductions*, are polynomial-time isomorphic, i.e., there exist bijective polynomial-time computable and polynomial-time invertible many-one reductions between them. They conjectured that this property holds for all NP-complete sets. Since polynomial-time isomorphic sets have polynomially related densities, and the NP-complete set SAT has exponential density, the Berman-Hartmanis conjecture implies that there are no sparse complete sets for NP under polynomial-time computable many-one reductions.

Intermediate types of reductions, such as bounded truth-table reductions, also play an important role.

Note that if P = NP, sparse complete sets for NP obviously exist under any reasonable polynomial-time computable reducibility. In this respect, Mahaney [22] settled the problem for many-one reductions: There is no sparse hard set for NP under polynomial-time many-one reductions unless P = NP. It appears very difficult to prove that the same holds for Turing reductions, but Karp and Lipton [18] showed that the existence of a sparse hard set for NP under polynomial-time Turing reductions collapses the polynomial-time hierarchy to the second level. In bridging the gap between many-one reductions and Turing reductions, Ogiwara and Watanabe [28] obtained a breakthrough by extending Mahaney's theorem to bounded truth-table reductions: There is no sparse hard set for NP under polynomial-time bounded truth-table reductions unless P = NP.

Given this state of affairs, complexity theorists wondered whether randomization might help: They started investigating the existence of sparse hard sets for NP under randomized polynomial-time reductions.

Cai, Naik and Sivakumar [7] showed that there is no sparse hard set for NP under polynomial-time randomized two-sided error many-one reductions with confidence as small as inversely polynomial unless NP = RP. We prove as one of our results that the same holds for bounded truth-table reductions.

1.2 The Sparse Hard Set Problem for P

This paper mainly deals with the sparse hard set problem for P: Are there sparse hard sets for P under logspace computable reducibilities? Similar considerations as for the sparse hard set problem for NP motivate this question:

- The sets that reduce to sparse sets under *Turing reductions* computable by logspace-uniform log-depth circuits are exactly those that have (non-uniform) polynomial-size log-depth circuits, and the same holds for truth-table reductions instead of Turing reductions.
- Based on the observation that all known sets complete for P under logspace computable *many-one reductions*, are in fact logspace isomorphic, Hartmanis [14] conjectured that all P-complete sets are logspace isomorphic. By the same token as above, since a P-complete set of exponential density exists, this implies that sets complete for P under logspace computable many-one reductions cannot be sparse.

As before, note that if P = L, sparse complete sets for P clearly exist under any reasonable logspace computable reducibility. Recently, Ogihara [26] made significant progress in showing this condition also necessary in the case of many-one reductions, and Cai and Sivakumar [8] actually proved it: There is no sparse hard set for P under logspace many-one reductions unless P = L. Cai, Naik and Sivakumar [7] then tried to extend this result to bounded truth-table reductions, and we establish this extension here: There is no sparse hard set for P under logspace computable bounded truth-table reductions unless P = L. We generalize this theorem and obtain new results for various sparseness conditions, space bounds and bounds on the number of queries of the truth-table reduction. However, the problem remains open for general logspace truth-table reductions, which are equivalent to logspace Turing reductions [20].

Regarding randomized reductions, Cai, Naik and Sivakumar [7] showed that there is no sparse hard set for P under logspace randomized two-sided error many-one reductions with confidence at least inversely polynomial unless all of P has randomized logspace algorithms with zero error, provided two-way read access to the random bit tape used. We establish that this theorem carries through for bounded truth-table reductions.

1.3 Overview of the Paper

We fix our notation and state some preliminaries in section 2.

Section 3 deals with deterministic reductions. We first describe previous work and indicate how our results relate to it.

Next we prove our Main Theorem that there is no sparse hard set for P under logspace computable bounded truth-table reductions unless P collapses to L. As in previous papers [26, 8, 7, 6], the proof structure parallels the one used by Ogiwara and Watanabe [28] for NP. We construct a space-efficient algorithm for the P-complete circuit value problem, based on the reduction of a well-chosen auxiliary set in P to a sparse set. We use the auxiliary set defined by Cai and Sivakumar [8], which encapsulates Reed-Solomon encodings [21]of the gate values of the circuits. The main ingredients of the algorithm are an NC^1 Vandermonde system solver [12] to recover the encoded gate values, and a novel combinatorial argument for exploiting sparseness. Similar to Cai and Sivakumar's construction, in case of reductions computable in NC^1, our algorithm for the circuit value problem is actually NC^1. We single out the crucial new way of exploiting the sparseness in a separate Combinatorial Lemma, and also consider some consequences of the Main Theorem for complexity classes above P.

Then we generalize our Main Theorem and obtain an analogue for P of Homer and Longpré's result [15] for NP. The generalization parameterizes the sparseness condition, the space bound and the bound on the number of queries of the truth-table reduction. Another instantiation of this Generic Theorem yields that that there is no quasipolynomially dense hard set for P under polylog-space computable truth-table reductions using polylogarithmically many queries unless P is in polylog-space.

Finally, we apply the proof technique to NL and L, as Cai and Sivakumar [9] did for their results. This shows that there is no sparse hard set for NL under logspace computable bounded truth-table reductions unless NL = L, and that there is no sparse hard set for L under NC^1-computable bounded truth-table reductions unless L = NC^1.

Section 4 presents our results on randomized reductions with two-sided error. We start off again with a description of earlier work.

Then we establish an analogue of our Main Theorem for such randomized reductions with confidence at least inversely polynomial: There is no sparse hard set for P under randomized two-sided error bounded truth-table reductions computable in logspace and with confidence at least inversely polynomial unless P is in randomized logspace, provided two-way read access to the random tape. This extends a result by Cai, Naik and Sivakumar [7] for many-one reductions. As in that paper, the proof uses the Hadamard encoding [21] of gate assignments of circuits defined by Ogihara [26] to construct an auxiliary set. Instead of the Vandermonde system solver in the deterministic case, we apply Goldreich and Levin's algorithm [13] to recover the encoded gate values, and combine it with a modified version of our Combinatorial Lemma.

Next we parameterize the Randomized Main Theorem in the same vein as the deterministic Main Theorem. The resulting Randomized Generic Theorem also yields that there is no quasipolynomially dense hard set for P under randomized two-sided error polylog-space computable truth-table reductions with confidence at least inversely quasipolynomial and making polylogarithmically many queries unless P is in randomized polylog-space, again using two-way read access to the random tape.

Finally, we apply the same technique to complexity classes other than P. As in the deterministic case, we obtain results for NL and L. Moreover, in the randomized case we can also apply our construction to NP by virtue of Valiant and Vazirani's reduction [32], as Cai, Naik and Sivakumar [7] showed for many-one reductions: There is no sparse hard set for NP under randomized two-sided error polynomial-time bounded truth-table reductions with confidence at least inversely polynomial unless NP = RP.

We conclude the paper by mentioning possible directions for further research in section 5.

2 Notation and Preliminaries

Most of our notation and definitions are standard [3].

Σ denotes the binary alphabet $\{0,1\}$. We consider the lexicographic ordering $\leqslant_\ell$ on Σ^*. An interval of a set $A \subseteq \Sigma^*$ is a subset of the form $\{w \in A \mid w_1 \leqslant_\ell w \leqslant_\ell w_2\}$ for some $w_1, w_2 \in A$. The set A is sparse if c_A is bounded by a polynomial, where $c_A : \mathbb{N} \rightarrow \mathbb{N} : n \rightarrow |A \cap \Sigma^{\leqslant n}|$.

We say that a set system $D : \Sigma^* \rightarrow 2^{\Sigma^*}$ can be generated in space $s(n)$ if there is a Turing machine that on input $y \in \Sigma^n$ enumerates the elements of $D(y)$ using work space $s(n)$.

Unless stated otherwise, all circuit families are supposed to be logspace uniform. In particular, NC^1 is the class of sets computable by logspace uniform circuits of logarithmic depth.

A Turing reduction of a set A to a set B is an oracle Turing machine M such that $M^B = A$. A truth-table reduction is a Turing reduction M whose queries are independent of the oracle B. If the number of queries M makes on any input is bounded, we call M a bounded truth-table reduction. If the reduction asks only one query and answers the answer to that query, it is called a many-one reduction.

We use Wilson's generalization [34] of Cook's [10] concept of oracle circuits to define reductions computed by circuits. An oracle gate with m inputs contributes $\log m$ to the length of any path through it.

An instance of the circuit value problem, CVP for short, consists of the description of a circuit C with one marked gate and an input x, where C has $|x|$ inputs. The instance $\langle C, x\rangle$ is in CVP if the marked gate of C outputs 1 on input x. CVP is complete for P under logspace many-one reductions [19]. Hence, it suffices to prove that CVP is in L to establish that $\mathrm{P} \subseteq \mathrm{L}$. CVP is actually complete for P under weaker reductions, e.g., under NC^1-computable many-one reductions. So, it also suffices to prove that CVP is in NC^1 to establish that $\mathrm{P} \subseteq \mathrm{NC}^1$.

An instance of the s-t-connectivity problem for directed acyclic graphs, DAG-STCON for short, consists of the description of a directed acyclic graph G and two of its vertices s and t. The instance $\langle G, s, t\rangle$ belongs to DAG-STCON if there is a directed path in G from vertex s to vertex t. DAG-STCON is complete for NL under logspace many-one reductions [30] and actually even under NC^1-computable many-one reductions.

We denote the analogous problem for undirected acyclic graphs by F-STCON ("F" for forest). It is complete for L under NC^1-computable many-one reductions [11].

We use the following notation introduced by Nisan [25]: $\mathrm{RP}^*\mathrm{SPACE}[s(n)]$ denotes the class of sets A that can be accepted by a one-sided error probabilistic Turing machine M with two-way read access to its random tape that uses no more than $s(n)$ work space on inputs of length n and always halts. One-sided error means that M always rejects an input outside of A, and that it accepts an input in A with probability at least δ for some positive constant δ. $\mathrm{ZP}^*\mathrm{SPACE}[s(n)]$ is defined as $\mathrm{RP}^*\mathrm{SPACE}[s(n)] \cap \mathrm{coRP}^*\mathrm{SPACE}[s(n)]$. $\mathrm{R}^*\mathrm{L}$ is short for $\mathrm{RP}^*\mathrm{SPACE}[\log n]$, and $\mathrm{ZP}^*\mathrm{L}$ for $\mathrm{ZP}^*\mathrm{SPACE}[\log n]$.

A randomized two-sided error reduction of a set A has confidence δ if, for any input x, it answers the membership question "x to A?" correctly with probability at least $\frac{1}{2} + \delta$.

$\mathrm{GF}(2^b)$ denotes the finite field with 2^b elements ($b \in \mathbb{N}$, $b \geqslant 1$). It can be defined as the set of equivalence classes of polynomials over GF(2) modulo an irreducible polynomial $f(x)$ of degree b over GF(2). Consequently, its elements can be represented as polynomials of degree less than b over GF(2). For $b \in \{2 \cdot 3^t \mid t \in \mathbb{N}\}$, $f(x) = x^b + x^{b/2} + 1$ is irreducible over GF(2) [33], and we will use this polynomial to construct $\mathrm{GF}(2^b)$ for such b's.

3 Deterministic Reductions

We start off with an overview of previous work, and describe how our results evolve from it. The reader can skip this subsection without loss of continuity. Then we prove our Main Theorem, which we next parameterize to a Generic Theorem for P. Finally, we apply the same technique to the sparse hard set problem for NL and L.

3.1 Previous Work

The proofs of results on the sparse hard set problem for NP usually go by constructing in polynomial-time a set of assignments to the variables of a given formula that contains at least one satisfying assignment in case of a satisfiable formula. Since we can check the validity of an assignment in polynomial-time, this yields a polynomial-time algorithm for the NP-complete problem SAT, which shows that P = NP. For the sparse hard set problem for P, we can use a similar approach based on the P-completeness of CVP. In this case we use gate assignments as membership witnesses which we can check for validity in logspace. So, using the reduction of some set(s) A in P to a sparse set S, we have to construct within the given resource bounds a set of gate assignments which contains the correct one for the given input. We will call such a set a witness set.

Much of the progress on this kind of problems relies on an appropriate choice of A. Ogihara [26] considered the set

$$A_1 = \{\langle C, x, y, a\rangle \mid C \text{ circuit with } |x| \text{ inputs}, x \in \Sigma^*, y \in \Sigma^m, a \in \Sigma \text{ and } \oplus_{y_j=1} g_j = a\}, \tag{1}$$

where m is the number of gates of C, $g_1, \ldots, g_m$ denote the gate values of C on input x, and $\oplus$ symbols exclusive or. This set is in P and has the following simple, but interesting properties: Let f be a many-one reduction of A_1 to a set S.

Property 3.1 *If $f(\langle C, x, y, a\rangle) \in S$, then*

$$\bigoplus_{y_j=1} g_j = a. \tag{2}$$

Property 3.2 *For any syntactically correct $\langle C, x, y\rangle$, one of $f(\langle C, x, y, 0\rangle)$ and $f(\langle C, x, y, 1\rangle)$ is in S.*

If S is sparse, for any fixed instance $\langle C, x\rangle$ of CVP, Property 3.2 guarantees the existence of a string $w = f(\langle C, x, y^*, a^*\rangle) \in S$ to which many $(y, a) \in \Sigma^m \times \Sigma$ are mapped by f, so that according to Property 3.1

$$\bigoplus_{y_j=1} g_j = \bigoplus_{y_j^*=1} g_j \oplus a^* \oplus a. \tag{3}$$

This equation actually also holds if $w \notin S$. In any case, if $y \neq y^*$, it allows to express one of the gate values as the parity of some other gate values.

Based on this observation, Ogihara managed to transform in DSPACE[$\log^2 n$] each of the gates of the given circuit C into parity gates with the same values for the given input x, except for an $O(\log n)$ sized subset G of the gates. Since the circuit value problem for parity circuits is in DSPACE[$\log^2 n$], it suffices to cycle through all possible assignments to the $O(\log n)$ gates in G to construct a DSPACE[$\log^2 n$] enumerable witness set for CVP. A DSPACE[$\log^2 n$] algorithm for CVP follows.

Cai and Sivakumar [8] also used the set A_1, but viewed (2) as the linear equation

$$\sum_{j=1}^{m} y_j g_j = a \tag{4}$$

over GF(2) in the gate values $g_1, \ldots, g_m$. Their algorithm tried to set up a system of such equations of rank $m - O(\log m)$, which allowed them to obtain a witness set for CVP in NC^2 based on the following facts:

Fact 3.1 *The rank of a matrix can be determined in arithmetic* NC^2 *[23].*

Fact 3.2 *A square system of linear equations of full rank can be solved in arithmetic* NC^2 *[5].*

Note that when working over GF(2), arithmetic NC^2 is equivalent to its Boolean counterpart.

Using Fact 3.1, given a system of equations of rank $m - O(\log m)$ over GF(2) in the gate values, Cai and Sivakumar construct in NC^2 a subset of these equations and an $O(\log m)$ sized subset G of the gates, such that the resulting system in the other gate values is square and of full rank. The application of Fact 3.2 to these systems for all possible assignments to the $O(\log m)$ gates in G in parallel, results in an NC^2 witness set for CVP. They show that the system consisting of all equations over GF(2) corresponding to (3) for $(y, a), (y^*, a^*) \in Y \subseteq \Sigma^m \times \Sigma$ for which $f(\langle C, x, y, a\rangle) = f(\langle C, x, y^*, a^*\rangle)$, has rank $m - O(\log m)$ with high probability, if Y is a polynomial sized uniform sample from $\Sigma^m \times \Sigma$. This yields an RNC^2 algorithm for CVP.

Next they prove that the rank of this system is always $m - O(\log m)$, if $Y = D \times \Sigma$ with D the following "small-bias sample space" [24, 1]:

$$D = \{(\prec u^i, v \succ)_{i=0}^{m-1} \mid u, v \in \mathrm{GF}(2^b)\}, \tag{5}$$

where $b \in \{2 \cdot 3^t \mid t \in \mathbb{N}\}$ with $b \in O(\log m)$, and $\prec \cdot, \cdot \succ$ represents the standard inner product of the vector space $\mathrm{GF}(2^b)$ over GF(2). This way, they effectively derandomize their RNC^2 algorithm to obtain an NC^2 procedure for CVP.

Finally, probably inspired by the construction of the sample space D used in the derandomization, Cai and Sivakumar considered the set

$$\begin{aligned} A_2 = \{\langle C, x, u, v\rangle \mid\ & C \text{ circuit with } |x| \text{ inputs}, x \in \Sigma^*, \\ & u, v \in \mathrm{GF}(2^b) \text{ for some } b \in \{2 \cdot 3^t \mid t \in \mathbb{N}\} \text{ and } \sum_{j=1}^{m} u^{j-1} g_j = v\}, \end{aligned} \tag{6}$$

where m again denotes the number of gates of C, and $g_1, \ldots, g_m$ the gate values of C on input x. This set also is in P and has the following properties similar to A_1: Let f be a many-one reduction of A_2 to a set S.

Property 3.3 *If* $f(\langle C, x, u, v\rangle) \in S$, *then*

$$\sum_{j=1}^{m} u^{j-1} g_j = v. \tag{7}$$

Property 3.4 *For any syntactically correct* $\langle C, x, u\rangle$ *where* $u \in \mathrm{GF}(2^b)$, *there is (exactly) one* $v \in \mathrm{GF}(2^b)$ *such that* $f(\langle C, x, u, v\rangle) \in S$.

For fixed $\langle C, x\rangle$, we say that $\langle u, v\rangle$ generates equation (7). The equations that result from Property 3.3 form a Vandermonde system, which is particularly interesting because we can determine the rank of and solve Vandermonde systems in NC^1 [12]. If b is sufficiently large, Property 3.4 guarantees that for at least one pair $(u^*, v^*) \in \mathrm{GF}(2^b) \times \mathrm{GF}(2^b)$ the set

$$E(u^*, v^*) \doteq \{(u, v) \in \mathrm{GF}(2^b) \times \mathrm{GF}(2^b) \mid f(\langle C, x, u, v\rangle) = w\}$$

contains at least m pairs, where $w \doteq f(\langle C, x, u^*, v^*\rangle) \in S$. Hence, it contains only generators of correct equations, and these equations form a Vandermonde system of full column rank m. If S is sparse, $b \in \Omega(\log m)$ suffices. So, the solutions to an $m \times m$ system generated by $E(u^*, v^*)$ for all possible pairs $(u^*, v^*) \in \mathrm{GF}(2^b) \times \mathrm{GF}(2^b)$ form a witness set for CVP with complexity NC^1 modulo the complexity of the reduction, and complexity L if we take the logspace complexity of the reduction into account. The corresponding circuit for CVP is sketched in Figure 1, where E_j denotes the set $E(u^*, v^*)$ for the j-th pair (u^*, v^*). This way, Cai and Sivakumar proved that the existence of a sparse hard set for P under logspace many-one reductions implies that CVP $\in$ L. We will use the same scheme to prove that proposition for logspace bounded truth-table reductions, that is: Construct a logspace algorithm for CVP with the structure outlined in Figure 1 using a logspace bounded truth-table reduction of A_2 to a sparse set S.

When assuming the existence of a k-truth-table reduction of A_1 or A_2 to a set S, there is no analogue of Properties 2 and 4, i.e., we cannot guarantee that many of the queries belong to S, and hence, if S is sparse, many map to the same query (in S). However, for the case of a reduction from A_1, Cai, Naik and Sivakumar [7] made (a slightly weaker version of) the following observation: Let Y be any subset of $\Sigma^m \times \Sigma$.

- Either there is a popular query, i.e., a string w that the reduction queries for many inputs $\langle C, x, y, a\rangle$ where (y, a) ranges over Y. In that case, we can reduce the problem to one involving a $(k-1)$-truth-table reduction by restricting the inputs to the ones for which the reduction queries w, and assuming both $w \in S$ and $w \notin S$ in parallel. Since many pairs (y, a) in Y map to w, this restriction does not decrease the number of pairs (y, a) by too much, and in one of the parallel executions, the assumption we make is correct.

- Either there are no popular queries, i.e., for any string w there are no more than say p pairs $(y, a) \in Y$ for which the reduction queries w on input $\langle C, x, y, a\rangle$. If s is a bound on the length of the queries, this implies that there are at most $c_S(s) \cdot p$ pairs (y, a) in Y for which some queries belong to S. Hence, if we partition Y into $c_S(s) \cdot p + 1$ classes of equal size ($\pm$ 1), then for at least one class the assumption that all queries made are outside of S is correct, and that class contains at least $\lfloor \frac{|Y|}{c_S(s)\cdot p+1} \rfloor$ pairs. All classes can be processed in parallel, and provided $c_S(s) \cdot p$ is not too large compared to $|Y|$, once again, we end up with large subclasses Y' of Y for at least one of which we make correct assumptions.

Cai, Naik and Sivakumar then showed that for a sufficiently large uniform sample Y of $\Sigma^m \times \Sigma$, with high probability, for every large subset Y' of Y the system of equations

$$\sum_{j=1}^{m} y_j g_j = a + \chi_{A_1}(\langle C, x, y, a\rangle) + 1, \tag{8}$$

where (y, a) ranges over Y', has high rank. Note that equation (8) is always correct, even if $\langle C, x, y, a\rangle \notin A_1$, because we are working over GF(2). We can compute the right-hand side of (8) using our assumptions about the membership to S of the queries, provided these assumptions are correct. By the above procedure we can construct in NC^1 a class of large subsets Y' for at least one of which all our assumptions about the membership queries are right. Cai, Naik and Sivakumar proved that a polynomial sized sample Y suffices to obtain rank $m - O(\log m)$ with high probability, so the same strategy as discussed above for many-one reductions allowed them to solve CVP in RNC^2.

If we want to apply this idea using a bounded truth-table reduction from A_2 instead of from A_1, there is an additional difficulty related to the fact that we are no longer computing over GF(2). In the case of A_1, for every (y, a) for which the memberships to S of the truth-table queries are known, we can construct a correct equation (8), even if the reduction rejects. For A_2 on the other hand, this is only the case for generators (u, v) of *correct* equations, i.e., when the reduction accepts, and this only happens for a small fraction of the generators. The reduction rejecting tells us that our guess for the right-hand side is wrong, but unlike when working over GF(2), that knowledge does not suffice to deduce the correct value of the right-hand side.

So, we have to restrict our attention to the set of generators (u, v) for which the reduction accepts. If there is a query that is popular among generators of correct equations, we can reduce the problem as above. However, there is a complication in the other case. In considering only the set G of generators for which the reduction accepts assuming all queries outside of S, we exclude at most $c_S(s) \cdot p$ of the generators of correct equations, so that should not be a problem. But there can be a lot of generators of incorrect equations in G for which not all queries are outside of S, because the bound p only holds for generators of correct equations. Therefore, we cannot tightly bound the number of generators in G for which we erroneously assume the reduction only queries strings outside of S. So, we cannot use the idea of partitioning G into subsets of equal size, at least one of which only contains generators all of whose queries are outside of S, since the resulting subsets would have to be too small. We will solve this problem by using variable sized subsets of a particular kind obtained by considering *intervals* of allowable queries.

For the sake of completeness of the overview, we mention the result by Cai, Naik and Sivakumar [7, 6] that the existence of a sparse hard set for P under logspace bounded truth-table reductions implies that $\mathrm{P} \subseteq \mathrm{NC}^2$. They use a reduction from A_1 for all elements of the set D defined by equation (5) and certain error-correcting capabilities of this small-bias sample space. This allows them to distill out of the equations (4) for all $(y, a) \in D \times \Sigma$ for which the reduction accepts assuming all queries outside of S, a correct full rank Vandermonde system over $\mathrm{GF}(2^b)$ in the gate values, provided the assumption does not introduce too many false equations of the form (4). They show that if it does, the NC^2 approach of Cai and Sivakumar [8] works. However, our work does not build upon this construction.

3.2 Main Result

In this subsection, we establish our Main Theorem. The proof will rely on a Combinatorial Lemma indicating how in general a space efficient reduction to a sparse set can be exploited, and on a known algorithm to solve Vandermonde systems of equations [12]. We will first prove the Main Theorem, and then the Combinatorial Lemma. Finally, we will mention some consequences of the Main Theorem for complexity classes above P.

3.2.1 Proof of the Main Theorem

Assuming the existence of a sparse hard set for P under logspace bounded truth-table reductions, we will show how to decide in logspace the P-complete set CVP.

A useful aspect of CVP in this context is that it has natural logspace verifiable membership proofs: Gate assignments can be checked for validity in logspace. One way of solving CVP consists of cycling through all possible gate assignments and checking each of them for validity. In general this approach does not yield a logspace algorithm, because generating all gate assignments requires linear space, but the bounded truth-table reducibility of P to a sparse set will allow us to reduce the search space to a subset we can generate in logspace.

In order to do so, we consider the auxiliary set A_2 defined by Cai and Sivakumar [8]:

$$A_2 = \{\langle C, x, u, v\rangle \,|\, C \text{ circuit with } |x| \text{ inputs, }, x \in \Sigma^*, \\ u, v \in \mathrm{GF}(2^b) \text{ for some } b \in \{2 \cdot 3^t \,|\, t \in \mathbb{N}\} \text{ and } \sum_{j=1}^{m} u^{j-1} g_j = v\}, \tag{9}$$

where $g_1, \ldots, g_m$ denote the gate values of C on input x. For any instance $y \doteq \langle C, x\rangle$ of CVP with m gates, and any $b \in \{2{\cdot}3^t \,|\, t \in \mathbb{N}\}$, A_2 contains a Reed-Solomon encoding [21] of the gate assignment of C on input x, which has the following property: The knowledge of m pairs $(u, v) \in \mathrm{GF}(2^b) \times \mathrm{GF}(2^b)$ such that $\langle C, x, u, v\rangle \in A_2$ enables us to recover the gate assignment as the solution of the $m \times m$ Vandermonde system $\Sigma_{j=1}^{m} u^{j-1} g_j = v$. We can solve such systems in space $O(b + \log m)$ using the algorithm described by Eberly [12].

Lemma 3.1 *Given an irreducible polynomial $f(x)$ of degree b over* $\mathrm{GF}(2)$ *defining* $\mathrm{GF}(2^b)$*, an $n \times n$ full rank Vandermonde system over* $\mathrm{GF}(2^b)$ *can be solved by* $\mathrm{DSPACE}[b + \log n]$ *uniform circuits of depth $O(b + \log n)$.*

Note that we use $f(x) = x^b + x^{b/2} + 1$ as the irreducible polynomial over GF(2) defining $\mathrm{GF}(2^b)$.

So, we are done if, for a given instance y of CVP, we can generate in logspace a collection of sets at least one of which is a subset of $A_2 \cap D(y)$ of size at least m, where $D(y) \doteq \{\langle y, u, v\rangle | u, v \in \mathrm{GF}(2^{b(y)})\}$ for some function $b : \Sigma^* \to \mathbb{N}$ such that $b(y) \in O(\log |y|)$. Observe that $|A_2 \cap D(y)| = 2^{b(y)}$.

The following Combinatorial Lemma allows us to do that. We will prove the lemma in Section 3.2.2.

Lemma 3.2 (Combinatorial Lemma) *Let A be logspace k-truth-table reducible to a sparse set S for some $k \in \mathbb{N}$, and let $D : \Sigma^* \to 2^{\Sigma^*}$ be a set system that can be generated in logspace. Suppose that the reduction queries strings of length at most $s(n)$ on inputs of $D(y)$ for $y \in \Sigma^n$. Then there is a logspace algorithm that on input $y \in \Sigma^n$ generates a collection of subsets of $D(y)$ at least one of which is a subset of A of size at least*

$$\frac{|A \cap D(y)|}{e \cdot (c_S(s(n)) + 1)^k}, \tag{10}$$

where e denotes the base of the natural logarithm.

We can apply this lemma to the set A_2 defined by (9), since A_2 belongs to P, whence by hypothesis logspace k-truth-table reduces to a sparse set S for some $k \in \mathbb{N}$. To prove A_2 in P, we use the fact that we constructed $\mathrm{GF}(2^b)$ based on the explicit irreducible polynomial $f(x) = x^b + x^{b/2} + 1$, which allows us to perform all arithmetic in polynomial-time. There are functions $b : \Sigma^* \to \{2 \cdot 3^t \,|\, t \in \mathbb{N}\}$ and $s : \mathbb{N} \to \mathbb{N}$ such that $|A_2 \cap D(y)| = 2^{b(y)} \geqslant n \cdot e \cdot (c_S(s(n)) + 1)^k$ and such that the reduction of A_2 to S queries strings of length at most $s(n)$ on inputs of $D(y)$ for $y \in \Sigma^n$. Moreover, we can choose $b(y) \in O(\log |y|)$ and space constructible, so that we can generate D in logspace. Since the number of gates is no more than n, the Combinatorial Lemma yields a collection of sets with the properties we need. ☐

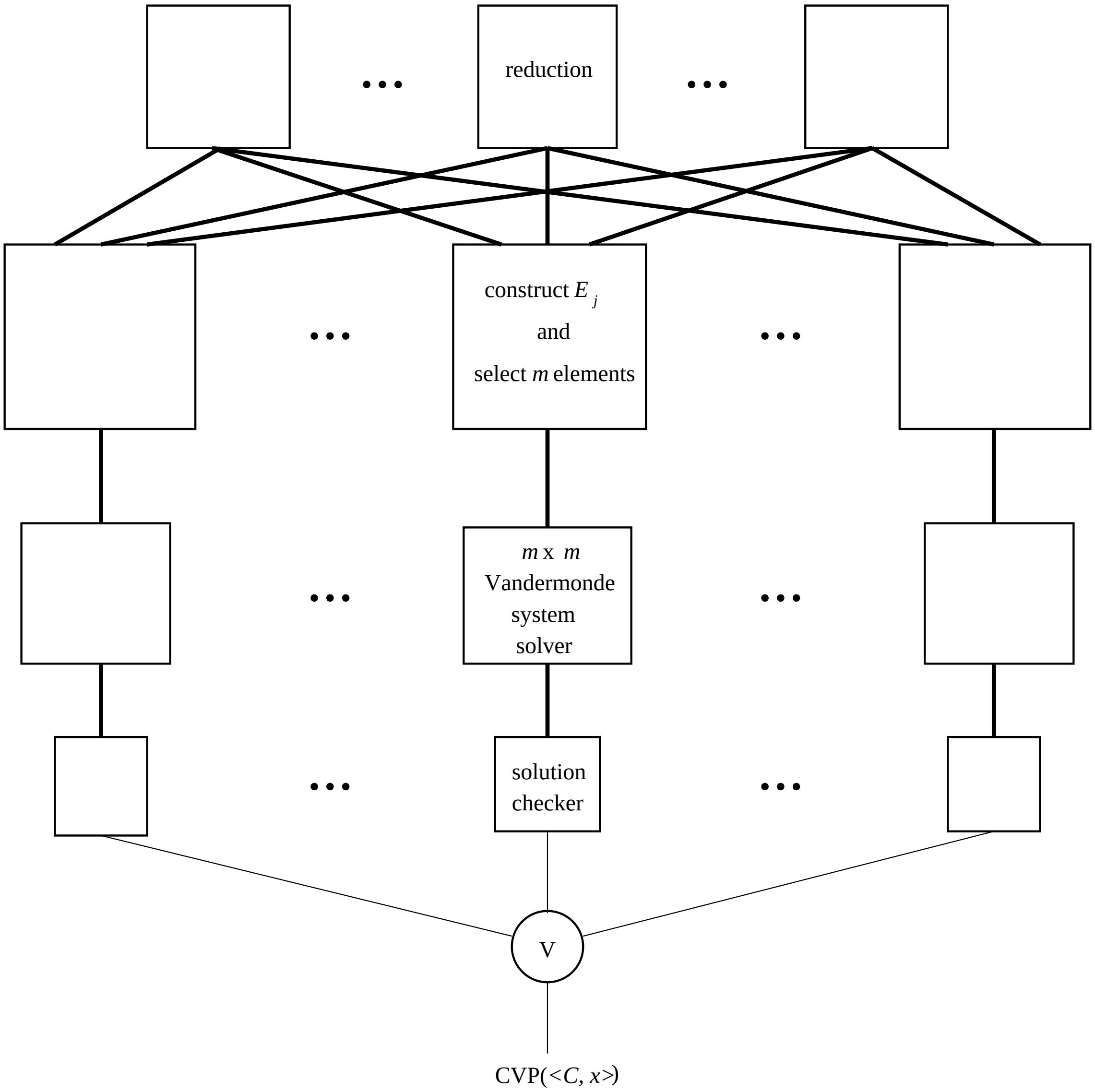

Figure 1: Structure of an algorithm for the circuit value problem.

The structure of the resulting algorithm for CVP is outlined in Figure 1, where E_j denotes the j-th set generated by the Combinatorial Lemma. Except for the reductions, we can actually implement every component of Figure 1 in NC^1, implying that our algorithm for CVP is NC^1 modulo the complexity of the reduction. Since proving that CVP is in NC^1 suffices to show that $\mathrm{P} = \mathrm{NC}^1$, we obtain the following:

Theorem 3.3 *There is no sparse hard set for* P *under* NC^1*-computable bounded truth-table reductions unless* $\mathrm{P} = \mathrm{NC}^1$.

3.2.2 Proof of the Combinatorial Lemma

Suppose A logspace k-truth-table reduces to a sparse set S for some $k \in \mathbb{N}$. Drawing meaningful conclusions about membership to A based only on a polynomial upper bound on the density of S, seems to require the application of the reduction to a sufficiently (polynomially) large set D of inputs with polynomially related lengths.

If we do that for a set $D \doteq D(y)$ of a set system which we can generate in logspace, and $A \cap D$ is large, we will be able to generate in logspace a collection of subsets of D at least one of which is a large subset of A. Basically, in order to obtain that subset, we will cycle through elements of D and try to use the logspace reduction to determine membership of these elements to A. The only problem is that we do not know the membership to S of the queries the reduction makes. Therefore, we will try to find a large subset of $A \cap D$ for which we can guess the membership to S of the queries the reduction makes on its elements. More specifically, we will consider a set of RA-pairs, i.e., pairs of restrictions of D and corresponding assumptions about membership to S of the queries the reduction makes on inputs from D satisfying the restriction, such that the following conditions hold:

1. Given an RA-pair, we can check in logspace whether a given element of D satisfies the restriction, and determine the membership bits of the queries made by the reduction on that element, as implied by the assumption.

2. We can generate in logspace the set of all RA-pairs.

3. For at least one RA-pair, the membership bits implied by the assumption are correct for all inputs from D satisfying the restriction, and many inputs from $A \cap D$ satisfy the restriction.

Clearly, we reach our goal if we manage to meet these conditions: Cycle through the set of RA-pairs, and for each of them, output the corresponding subset of D by cycling through the elements of D and checking whether it satisfies the restriction and whether the reduction accepts it using the membership bits implied by the assumption.

In order to obtain such a collection of RA-pairs, consider the set $Q(D)$ of all queries the reduction makes on inputs belonging to D. The set S partitions $Q(D) \backslash S$ into a polynomial number of intervals. See Figure 2.

If none of the strings in S is queried for many inputs in $A \cap D$, then, S being sparse and $A \cap D$ large, a lot of inputs in $A \cap D$ must have all of their queries outside of S. Since each of these inputs maps to at most k intervals of $Q(D) \setminus S$ and the number of k-subsets of intervals is polynomial, at least one subset of k intervals must contain in its union all queries of a polynomial fraction of those inputs in $A \cap D$ that have all their queries outside of S. So, the restriction of D to inputs

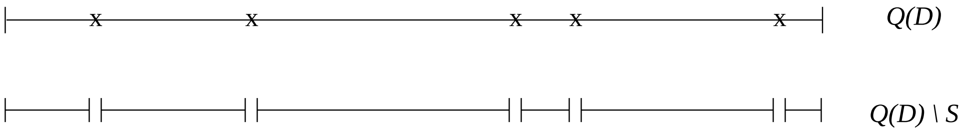

Figure 2: How S partitions $Q(D) \setminus S$.

for which the reduction only queries strings in the union of these k intervals, together with the assumption that all queries are outside of S, forms an RA-pair satisfying condition 3 (using an appropriate interpretation of "many" in this condition). The set of RA-pairs of this type, ranging over all k-subsets of intervals of $Q(D)$, clearly meets condition 1, and also condition 2, because we can generate the intervals of $Q(D)$ in logspace by cycling through all pairs of points of $Q(D)$.

In case there are popular queries in S, i.e., some strings in S are queried for many inputs in $A \cap D$, we can first restrict our attention to the inputs in D for which the reduction queries all strings in a maximal set P of jointly popular queries in S. The queries in P being jointly popular, this restriction does not reduce the size of $A \cap D$ by too much (second part of condition 3). Then we are left with a $(k - |P|)$-truth-table reduction to S without popular queries, so we can apply the procedure described in the previous paragraph. Since a set P of jointly popular queries necessarily is a subset of the queries made on some input of D, we can generate all possible P's in logspace by cycling through all points in D and, for each of them, through all (nonempty) subsets of the queries the reduction makes. So, with the associated assumption that all queries in P are in S, condition 2 is met. Conditions 1 and 3 are clearly also satisfied.

The resulting combined RA-pairs we consider are parameterized by a (possibly empty) subset P of queries the reduction makes on some input of D, and a subset of $k - |P|$ intervals $I_1, \ldots, I_{k-|P|}$ of $Q(D)$. The corresponding restriction G' of D is

$$G' = \{x \in D | \, P \subseteq Q(x) \text{ and } Q(x) \setminus P \subseteq \bigcup_{i=1}^{k-|P|} I_i\}, \tag{11}$$

and the associated assumption is that $Q(x) \cap S = P$ for $x \in G'$. The subset G of D generated by this RA-pair is

$$G = \{x \in D | \, P \subseteq Q(x) \text{ and } Q(x) \setminus P \subseteq \bigcup_{i=1}^{k-|P|} I_i \text{ and the reduction accepts } x \text{ on } P\},$$

where by "the reduction accepts x on P" we mean that it accepts x when using P instead of S to answer membership queries about S. It follows from the foregoing discussion that these RA-pairs satisfy all 3 conditions we needed.

Quantifying this argument yields the:

Proof (of the Combinatorial Lemma)
For any $n \in \mathbb{N}$ and $y \in \Sigma^n$, we will establish the existence of a set G' of type (11) with $P \subseteq S$ and the intervals I_i disjoint from S, for which $|A \cap G'|$ has the required size (10). We already argued that this suffices to prove the lemma.

Let $Q(x)$ denote the set of queries that the reduction makes on input x, D denote $D(y)$, $Q(D)$ the set of all queries on inputs $x \in D$, and d the bound (10).

Given any function $p : \{0, \dots, k+1\} \to [0, \infty)$, it is obvious that there is a set $P \subseteq S$ such that:

$$|\{x \in A \cap D |\, P \subseteq Q(x)\}| \geqslant p(|P|) \tag{12}$$
$$\forall\, w \in S \setminus P : |\{x \in A \cap D |\, P \cup \{w\} \subseteq Q(x)\}| \leqslant p(|P|+1), \tag{13}$$

provided that $p(0) \leqslant |A \cap D|$ and we set $p(k+1) = 0$. The function p describes a popularity criterion, and the set P represents a maximal subset of jointly popular queries in S with respect to p. We will determine p later on, once we know all properties we need of it, and it will become clear then why we allow p to have real non-integral values (because d is non-integral). Note that the range of $|P|$ is $\{0, \dots, k\}$.

Consider the set of inputs $G'' = \{x \in A \cap D |\, Q(x) \cap S = P\}$.

Claim 3.1 $|G''| \geqslant p(|P|) - c_S(s(n)) \cdot p(|P|+1)$.

Indeed, the set G'' contains all elements of the set on the left-hand side of (12), except for those that have at least one query in $S \setminus P$. Because of (13) the number of exceptions is bounded by $c_S(s(n)) \cdot p(|P|+1)$.

Claim 3.2 *There are intervals $I_1, \dots, I_{k-|P|}$ of $Q(D) \setminus S$ such that:*

$$|\{x \in G'' |\, Q(x) \setminus P \subseteq \bigcup_{i=1}^{k-|P|} I_i\}| \geqslant \frac{|G''|}{\max(1, \binom{c_S(s(n))+1}{k-|P|})}.$$

This is because S partitions $Q(D) \setminus S$ in at most $c_S(s(n)) + 1$ intervals (see Figure 2), and for each $x \in G''$, $Q(x) \setminus P$ is in the union of at most $k - |P|$ of these intervals.

The combination of Claims 1 and 2 yields that the set G' defined by (11) satisfies

$$|A \cap G'| \geqslant \frac{p(|P|) - c_S(s(n)) \cdot p(|P|+1)}{\max(1, \binom{c_S(s(n))+1}{k-|P|})},$$

which is at least d, provided that for $i \in \{0, \dots, k\}$

$$\frac{p(i) - c_S(s(n)) \cdot p(i+1)}{\max(1, \binom{c_S(s(n))+1}{k-i})} \geqslant d.$$

It is straightforward to check that the function

$$p(i) = d \cdot \sum_{j=0}^{k-i} (c_S(s(n)) + 1)^{k-i-j} \binom{c_S(s(n)) + 1}{j}$$

satisfies these conditions. The upper bound for $p(0)$ is also met, since $p(0) \leqslant d \cdot (c_S(s(n)) + 1)^k \cdot \sum_{j=0}^{k} \frac{1}{j!} \leqslant |A \cap D|$. The existence of p concludes the proof of the lemma. $\square$

3.2.3 Consequences of the Main Theorem for classes above P

Ogiwara and Watanabe [28] and Ogiwara and Lozano [27] proved that for $\mathcal{C}$ equal to NP, PP, $\mathrm{C}_=\mathrm{P}$, $\mathrm{Mod}_k\mathrm{P}$ for some $k \geqslant 2$, or PSPACE, there is no sparse hard set for $\mathcal{C}$ under polynomial-time bounded truth-table reductions unless $\mathcal{C} = \mathrm{P}$. Combining this result with Theorems 1.1 and 3.3 leads to the following observations:

Corollary 3.4 *Let $\mathcal{C}$ be* NP, PP, $\mathrm{C}_=\mathrm{P}$ *or* $\mathrm{Mod}_k\mathrm{P}$ *for some $k \geqslant 2$. There is no sparse hard set for $\mathcal{C}$ under logspace bounded truth-table reductions unless $\mathcal{C} = \mathrm{L}$.*

Corollary 3.5 *There is no sparse hard set for* PSPACE *under logspace bounded truth-table reductions.*

Corollary 3.6 *Let $\mathcal{C}$ be* NP, PP, $\mathrm{C}_=\mathrm{P}$ *or* $\mathrm{Mod}_k\mathrm{P}$ *for some $k \geqslant 2$. There is no sparse hard set for $\mathcal{C}$ under NC^1-computable bounded truth-table reductions unless $\mathcal{C} = \mathrm{NC}^1$.*

3.3 Generic Theorem for P

In this subsection, we generalize our Main Theorem to a Generic Theorem by parameterizing the sparseness condition, the space bound and the bound on the number of truth-table queries of the reduction. We first state and prove the Generic Combinatorial Lemma and the Generic Theorem, and then consider some interesting instantiations other than the Main Theorem.

3.3.1 Statement and Proof

The Combinatorial Lemma readily parameterizes to:

Lemma 3.7 (Generic Combinatorial Lemma) *Let A be truth-table reducible to S, and $D : \Sigma^* \to 2^{\Sigma^*}$ be a set system that can be generated in space $b(n)$. Suppose that the reduction queries at most $k(n)$ strings of length at most $s(n)$ and uses at most $a(n)$ space on inputs of $D(y)$ for $y \in \Sigma^n$. Then there is an algorithm running in space $O(k(n) \cdot (a(n) + b(n)))$ that on input $y \in \Sigma^n$ generates a collection of subsets of $D(y)$ at least one of which is a subset of A of size at least*

$$\frac{|A \cap D(y)|}{e \cdot (c_S(s(n)) + 1)^{k(n)}},$$

where e denotes the base of the natural logarithm.

Proof
We use the same algorithm as in the proof of the Combinatorial Lemma, i.e., for a given $y \in \Sigma^n$, we cycle through the set of RA-pairs associated with y, and for each of them, output the corresponding subset of $D(y)$ by cycling through the elements of $D(y)$ and checking whether it satisfies the restriction and whether the reduction accepts it using the membership bits implied by the assumption. The counting argument carries through and shows that the collection of subsets of $D(y)$ we generate contains at least one subset of A of the required size. So, we only have to check the space complexity of the algorithm.

Cycling through all RA-pairs amounts to:

- Cycling for all elements of $D(y)$ through all subsets of queries the reduction makes on that input. We can do this in $O(b(n)+k(n))$ space, namely $O(b(n))$ space to go through the list of elements of $D(y)$, and $k(n)$ or less bits to keep track of which queries belong to the subset.
- Cycling through $k(n)$ or less intervals of $Q(D(y))$ (see Lemma 3.2 for notation). For a single interval, it suffices to keep track of its endpoints, and we can represent them as the i-th query the reduction makes on some input of $D(y)$, which takes $O(b(n)+\log k(n)) \subseteq O(b(n)+a(n))$ space. So, this part has space complexity $O(k(n)\cdot(a(n)+b(n)))$.

With the given representations, we can check in $O(a(n))$ space whether an element of $D(y)$ satisfies the restriction, and whether the reduction accepts using the membership bits of the queries as implied by the assumption. It follows that the overall space complexity is $O(k(n)\cdot(a(n)+b(n)))$. $\square$

Using this lemma, we can generalize the Main Theorem to:

Theorem 3.8 (Generic Theorem) *There is no hard set for* P *of census at most $c(n)$ under truth-table reductions that query at most $O(k(n))$ strings of length at most $O(s(n))$ and run in space $O(a(n))$ on inputs of length n unless* $\mathrm{P} \subseteq \mathrm{DSPACE}[t(n^{O(1)})]$*, where*

$$\begin{aligned} t(n) &= k(n)\cdot(a(n)+b(n)) \\ b(n) &= \log n + k(n)\cdot \log c(s(n)), \end{aligned}$$

provided $b(n) \in o(n)$ and the following conditions hold: $b(n)$ is space constructible, $a(O(n)) \subseteq O(a(n))$, $\log c(O(s(O(n)))) \subseteq O(\log c(s(n)))$, and $k(O(n)) \subseteq O(k(n))$.

Proof
Since CVP is complete for P under logspace many-one reductions, it suffices to show that we can solve it in $\mathrm{DSPACE}[t(n)]$ assuming the existence of a set S of census at most $c(n)$ that is hard for P under truth-table reductions that query at most $O(k(n))$ strings of length at most $O(s(n))$ and run in space $O(a(n))$ on inputs of length n.

As in the proof of the Main Theorem, we will construct an algorithm for CVP with the structure of Figure 1: Using the Generic Combinatorial Lemma, we produce a collection of Vandermonde systems, at least one of which has the gate values as its unique solution. Solving these systems and checking the resulting gate assignments for validity then completes the algorithm.

We appy the Generic Combinatorial Lemma to the set A_2 defined by (9) and the set system $D(y) \doteq \{\langle y,u,v\rangle \mid u,v \in \mathrm{GF}(2^{b^*(y)})\}$, where $b^*: \Sigma^* \to \{2\cdot 3^t \mid t \in \mathbb{N}\}$ is such that $b^*(y) \approx \alpha \cdot b(|y|)$ for a sufficiently large constant α. It is clear that we can generate D in space $O(b(n))$. Let $y \in \Sigma^n$. Note that since $b^*(y) \in o(n)$, there is a constant β independent of α such that the the size of elements of $D(y)$ is asymptotically bounded by βn, and such that the reduction of A_2 to S queries at most $k^*(n) \leqslant \beta \cdot k(\beta n)$ strings of length at most $s^*(n) \leqslant \beta \cdot s(\beta n)$, and runs in space $a^*(n) \leqslant \beta \cdot a(\beta n)$ on inputs of $D(y)$. Under the given technical conditions, this implies that there is a constant γ independent of α such that asymptotically $k^*(n)\cdot\log(c(s^*(n))+1) \leqslant \gamma\cdot k(n)\cdot\log c(s(n)))$. Consequently,

$$\frac{|A_2 \cap D(y)|}{e\cdot(c_S(s^*(n))+1)^{k^*(n)}} \geqslant \frac{2^{b^*(y)}}{e\cdot(c(s^*(n))+1)^{k^*(n)}} \geqslant 2^{(\alpha-\gamma)b(n)-\log e}.$$

Since $b(n) \geqslant \log n$, the right-hand side is at least n for α sufficiently large. So, the Vandermonde system of correct equations we obtain, has full column rank, and we can recover the gate values by solving it.

Under the same technical conditions, the application of the Generic Combinatorial Lemma requires $O(t(n))$ space. Since we only need $O(b(n)) \subseteq O(t(n))$ space to solve the Vandermonde systems and check the resulting gate assignments for validity, our final algorithm for CVP has space complexity $O(t(n))$. □

3.3.2 Instantiations

For polynomially dense hard sets we obtain the following instantiation of the Generic Theorem, of which our Main Theorem is a special case.

Theorem 3.9 *Let $e \geqslant 1$ and $f \geqslant 0$. There is no sparse hard set for* P *under truth-table reductions computable in space $O(\log^e n)$ that make at most $O(\log^f n)$ queries unless* $\mathrm{P} \subseteq \mathrm{DSPACE}[\log^{e+2f} n]$.

Proof
Observing that $s(n) \in 2^{O(\log^e n)}$ in the Generic Theorem, immediately yields that $t(n) \in O(\log^{e+2f} n)$. □

For quasipolynomially dense hard sets we get:

Theorem 3.10 *Let $d > 0$, $e \geqslant 1$ and $f \geqslant 0$. There is no hard set for* P *with density bounded by $2^{O(\log^d n)}$ under truth-table reductions computable in space $O(\log^e n)$ that make at most $O(\log^f n)$ queries unless* $\mathrm{P} \subseteq \mathrm{DSPACE}[\log^a n]$, *where $a = \max(de + 2f, e + f)$.*

Proof
Observing that $s(n) \in 2^{O(\log^e n)}$ in the Generic Theorem yields that $b(n) \in O(\log n + \log^{de+f} n)$, and $t(n) \in O(\log^a n)$. □

In particular the following holds:

Corollary 3.11 *There is no quasipolynomially dense hard set for* P *under polylog-space truth-table reductions using no more than polylogarithmically many queries unless* P *is in polylog-space.*

3.4 Extension to classes other than P

In this subsection, we show how we can apply the idea of the Main Theorem to collapse a complexity class $\mathcal{C}_1$ other than P to a class $\mathcal{C}_2$ that contains NC^1, assuming the existence of a sparse hard set for $\mathcal{C}_1$ under bounded truth-table reductions computable with the power of $\mathcal{C}_2$. In order for the technique to be applicable, sets in $\mathcal{C}_1$ must have unique membership proofs that can be constructed in $\mathcal{C}_1$ and checked in $\mathcal{C}_2$. We can encode the membership proofs as solutions of Vandermonde systems and use the Combinatorial Lemma to recover them in a space efficient way using a space efficient bounded truth-table reduction of the encoding auxiliary set to a sparse set. This approach works for NL and L, as we will see now.

Theorem 3.12 *There is no sparse hard set for* NL *under logspace computable bounded truth-table reductions unless* $\mathrm{NL} = \mathrm{L}$.

Proof
Assuming the existence of a sparse hard set for NL under logspace bounded truth-table reductions, we will show how to solve in logspace the NL-complete problem DAG-STCON.

In order to do so, we consider a variant A_3 of the auxiliary set defined by Cai and Sivakumar [9]:

$$A_3 = \{\langle G, s, u, u^2, u^3, \ldots, u^{m-1}, v\rangle \,|\, G \text{ DAG with } m \text{ vertices, } s \text{ vertex of } G,$$
$$u, v \in \mathrm{GF}(2^b) \text{ for some } b \in \{2\cdot 3^t \,|\, t \in \mathbb{N}\} \text{ and } \sum_{j=1}^{m} u^{j-1} g_j = v\},$$

where g_j is a Boolean indicating whether G contains a directed path from s to the j-th vertex of G. Provided G is acyclic, the values $g_1, \ldots, g_m$ constitute a logspace verifiable proof of membership of $\langle G, s, t\rangle$ to DAG-STCON: We only have to check for every vertex $w \neq s$ of G that its g-value is 1 only if there an incoming edge from another vertex of G with g-value 1, and that vertex t has g-value 1. We can clearly perform these checks in logspace.

A_3 contains Reed-Solomon encodings of these values, just as A_2 in the proof of the Main Theorem. To check membership to A_3, in addition to some syntactical tests which clearly can be performed in NL, we have to do the following:

- Check that G is acyclic. We can do that in coNL, whence also in NL [17, 31].
- Check the successive powers of u for correctness. This involves verifying that the component of the input corresponding to u^j equals the product of u and the component of the input corresponding to u^{j-1} for $j = 2, \ldots, m-1$. We can do this in space $O(\log m + \log b)$, because we can compute the product of two elements of $\mathrm{GF}(2^b)$ in space $O(\log b)$ by computing the product of the corresponding polynomials of degree less than b over GF(2) and then reducing this product modulo $f(x) = x^b + x^{b/2} + 1$. To calculate the residue of a polynomial of degree less than $2b-1$ modulo $f(x)$ in space $O(\log b)$, we use the fact that
$$x^j \bmod f(x) = \begin{cases} x^{j-b/2} + x^{j-b} & \text{for } b \leqslant j < \frac{3}{2}b \\ x^{j-\frac{3}{2}b} & \text{for } \frac{3}{2}b \leqslant j < 2b. \end{cases}$$
- Check that the equation $\sum_{j=1}^{m} u^{j-1} g_j = v$ holds. We can do this in L^{NL}, whence in NL, since the g_j's can be computed within that complexity class, and once we have the successive powers of u, computing the left-hand side sum is easily performed in $O(\log m + \log b)$ space.

So, $A_3 \in$ NL, whence by hypothesis logspace k-truth-table reduces to the sparse set S for some $k \in \mathbb{N}$. Therefore, we can apply the same combination of the Combinatorial Lemma and the logspace Vandermonde system solver as in the proof of the Main Theorem to obtain a collection of values for $(g_j)_{j=1}^m$ which we can generate in logspace and such that at least one of them is correct. This yields a logspace algorithm for DAG-STCON with the structure of Figure 1. □

Once again, we can easily verify that the algorithm we construct in the above proof is actually NC^1 modulo the complexity of the reduction. Therefore, as DAG-STCON is complete for NL under NC^1-computable many-one reductions, we also obtain:

Theorem 3.13 *There is no sparse hard set for* NL *under* NC^1*-computable bounded truth-table reductions unless* $\mathrm{NL} = \mathrm{NC}^1$.

Regarding the sparse hard set problem for L, we can prove the following:

Theorem 3.14 *There is no sparse hard set for* L *under* NC^1*-computable bounded truth-table reductions unless* $\mathrm{L} = \mathrm{NC}^1$.

Proof
Assuming the existence of a sparse hard set for L under NC^1-computable bounded truth-table reductions, we will show how to solve in NC^1 the L-complete problem F-STCON.

Consider the auxiliary set

$$A_4 = \{\langle G, s, t, u, u^2, \ldots, u^{m-1}, v\rangle \mid G \text{ forest with } m \text{ edges, } s, t \text{ vertices of } G,$$
$$u, v \in \mathrm{GF}(2^b) \text{ for some } b \in \{2 \cdot 3^t \mid t \in \mathbb{N}\} \text{ and } \sum_{j=1}^{m} u^{j-1} e_j = v\},$$

where e_j is a Boolean indicating whether s and t belong to the same component of G and the j-th edge of G is on the unique path connecting s and t in G. Provided G is acyclic, the values $e_1, \ldots, e_m$ constitute an NC^1-verifiable proof of membership of $\langle G, s, t\rangle$ to F-STCON: It suffices to check that either $s = t$ or else in the subgraph of G defined by the e_j's with value 1, s and t have degree 1 and all other vertices have degree 0 or 2. We can clearly test this in NC^1.

A_4 contains Reed-Solomon encodings of the values $e_1, \ldots, e_m$ like A_3 in the proof of Theorem 3.12. Similar to that proof, to check membership to A_4, we basically have to do the following:

- Check that G is acyclic. Since this is equivalent to the membership of $\langle G, t, t\rangle$ to F-STCON, and F-STCON $\in$ L, we can perform this check in logspace [16].
- Check the successive powers of u for correctness. This can be done in logspace as argued in the proof of Theorem 3.12.
- Check that the equation $\sum_{j=1}^{m} u^{j-1} e_j = v$ holds. Since $e_j = 1$ iff $\langle G, s, t\rangle \in$ F-STCON and $\langle G - e_j, s, t\rangle \notin$ F-STCON, and F-STCON $\in L$, we can compute the e_j's in logspace. Using the powers of u given in the input, we can then compute the sum involved in space $O(\log m + \log b)$.

Therefore, $A_4 \in$ L, whence by hypothesis k-truth-table reduces to the sparse set S for some $k \in \mathbb{N}$ through a reduction computable in NC^1. So, once more, we can apply a combination of the Combinatorial Lemma and the NC^1 Vandermonde system solver as in the proof of the Main Theorem to obtain a collection of values for $(e_j)_{j=1}^m$ which we can generate in NC^1 and such that at least one of them is correct. This yields an NC^1 algorithm for F-STCON with the structure of Figure 1. □

4 Randomized Reductions

In this section, we describe our results on the existence of sparse hard sets under randomized bounded truth-table reductions with two-sided error, using the multiple access model of randomness, in which the Turing machine has two-way read access to a random bit tape. We start with a description of earlier work, which the reader can skip without loss of continuity. Then we prove an analogue of the Main Theorem for randomized two-sided error bounded truth-table reductions with confidence at least inversely polynomial. As in the deterministic case, we next parameterize it to a Randomized Generic Theorem for P. Finally, we prove similar results on the sparse hard set problem for NL, L, and also for NP.

4.1 Previous Work

Cai, Naik and Sivakumar [7] proved that there is no sparse hard set for P under randomized two-sided error many-one reductions computable in logspace and with confidence at least inversely polynomial unless $\mathrm{P} \subseteq \mathrm{ZP}^*\mathrm{L}$. They used Hadamard encodings [21] in the form of the auxiliary language A_1 defined by Ogihara [26], and a probabilistic algorithm by Goldreich and Levin [13] to recover the code word. The same approach will allow us to extend their result to bounded truth-table reductions.

Cai, Naik and Sivakumar applied the same technique to the sparse hard set problem for NL and L. Using the randomized reduction from satisfiability to unique satisfiability by Valiant and Vazirani [32], they were also able to obtain the first known result on the existence of sparse hard sets for NP under randomized many-one reductions with *two-sided* error: There is no sparse hard set for NP under polynomial-time randomized two-sided error many-one reductions with confidence at least inversely polynomial unless $\mathrm{NP} \subseteq \mathrm{RP}$. Earlier, Ranjan and Rohatgi [29] had shown this theorem for reductions with one-sided error on the non-membership side. Arvind, Köbler and Mundhenk [2] improved upon that, but the question whether the statement was true of reductions with one-sided error on the membership side, and of reductions with two-sided error remained open. Cai, Naik and Sivakumar answered that question positively, and we will show that the same holds for bounded truth-table reductions with two-sided error.

4.2 Main Result

In this section, we establish the analogue of our Main Theorem in the randomized setting:

Theorem 4.1 (Randomized Main Theorem) *There is no sparse hard set for* P *under randomized two-sided error bounded truth-table reductions computable in logspace and with confidence at least inversely polynomial unless* $\mathrm{P} \subseteq \mathrm{ZP}^*\mathrm{L}$.

The proof uses Hadamard encodings [21] instead of Reed-Solomon encodings as in the deterministic case. It is based on the probabilistic algorithm by Goldreich and Levin [13] to decode Hadamard codes, and on a randomized version of the Combinatorial Lemma. We first prove the Randomized Main Theorem, and then the Randomized Combinatorial Lemma.

4.2.1 Proof of the Randomized Main Theorem

Assuming the existence of a sparse hard set for P under randomized two-sided error bounded truth-table reductions computable in logspace and with confidence at least inversely polynomial, we will show that the P-complete set CVP is in $\mathrm{R}^*\mathrm{L}$. Since P is closed under complement, this suffices to prove that $\mathrm{P} \subseteq \mathrm{ZP}^*\mathrm{L} \doteq \mathrm{R}^*\mathrm{L} \cap \mathrm{coR}^*\mathrm{L}$.

As in the deterministic case, we will make use of gate assignments as logspace verifiable membership proofs for CVP. Using the randomized bounded truth-table reducibility of P to a sparse set, we will generate in $\mathrm{R}^*\mathrm{L}$ a list of gate assignments that contains at least one valid assignment, and check each of them for validity.

To create that list, we consider the auxiliary set A_5 defined by Ogihara [26]:

$$A_5 = \{\langle C, x, w\rangle \mid C \text{ circuit with } |x| \text{ inputs and } m \text{ gates, } x \in \Sigma^*,\ w \in (\mathrm{GF}(2))^m \text{ and } \sum_{j=1}^{m} w_j g_j = 1\}, \tag{14}$$

where g_j denotes the value of the j-th gate of C on input x, and the arithmetic is over GF(2). For any instance $y \doteq \langle C, x\rangle$ of CVP with m gates, $(\chi_A(\langle y, w\rangle))_{w\in(\mathrm{GF}(2))^m}$ is the Hadamard encoding of the gate values of C on input x. An interesting property of this encoding is that a randomized oracle with confidence at least inversely polynomial suffices to recover the code word in RNC^1. Goldreich and Levin [13] showed a way to do this.

Lemma 4.2 ([13]) *Let $\epsilon : \mathbb{N} \to (0, \frac{1}{2}]$, and suppose that $\ell(m) \doteq \log(\frac{2m}{\epsilon^2})$ is space constructible. Then there exists a* DSPACE$[\ell(m)]$ *uniform family of randomized oracle-augmented circuits $\{C_m\}_{m\in\mathbb{N}}$ of depth $O(\ell(m))$ making $(2^{\ell(m)}-1)m$ parallel oracle queries of length m, such that C_m on input 1^m outputs a list of $2^{\ell(m)}$ elements of Σ^m and the following holds: For any randomized Boolean oracle B on $\Sigma^*\times\Sigma^*$, for any $y\in\Sigma^*$, $m\in\mathbb{N}$, and $g\in\Sigma^m$, if*

$$\Pr[B(y,w) = g\cdot w] \geqslant \frac{1}{2} + \epsilon(m) \tag{15}$$

then

$$\Pr[\text{output of } C_m(1^m) \text{ with oracle } B(y,\cdot) \text{ contains } g] \geqslant \frac{1}{2},$$

where the former probability is with respect to the uniform distribution of w over Σ^m and the underlying distribution of $B(y,\cdot)$, the latter with respect to the uniform distribution over the random bits fed to C_m and the underlying distribution of $B(y,\cdot)$, and $g\cdot w$ denotes the inner product of g and w as vectors over GF(2).

So, we have to construct a randomized oracle B that can be computed by a logspace probabilistic Turing machine with two-way read access to its random tape, such that B approximates χ_A well in the following sense: For any instance y of CVP of length n with m gates

$$\Pr[B(y,w) = \chi_A(\langle y, w\rangle)] \geqslant \frac{1}{2} + \epsilon(m),$$

where $\epsilon(m)$ is at least inversely polynomial (so that $\ell(m)$ is logarithmic in m). The probability is with respect to the uniform distribution over Σ^m and the underlying probability measure of $B(y,\cdot)$. We cannot just use the reduction to construct B, because we do not know how to determine the membership to S of the queries in $\mathrm{R}^*\mathrm{L}$. However, it suffices to approximate χ_A well on a subset of instances of high probability. Indeed, suppose σ is a probabilistic predicate such that $\Pr[\sigma]$ is at least inversely polynomial in m, and π is a probabilistic predicate such that $\Pr[\pi = \chi_{A_5} \mid \sigma] - \frac{1}{2}$ is also at least inversely polynomial in m. Let B act as follows: If σ holds, then output π, otherwise output a random bit. Then

$$\begin{aligned}\Pr[B = \chi_{A_5}] &= \Pr[\sigma]\cdot\Pr[\pi = \chi_{A_5} \mid \sigma] + (1-\Pr[\sigma])\cdot\frac{1}{2}\\ &= \frac{1}{2} + \Pr[\sigma]\cdot(\Pr[\pi = \chi_{A_5} \mid \sigma] - \frac{1}{2}),\end{aligned}$$

which exceeds $\frac{1}{2}$ by at least the inverse of a polynomial in m. So, provided we can compute σ and π in $\mathrm{R}^*\mathrm{L}$, we are done.

The following Randomized Combinatorial Lemma yields the probabilistic predicates σ and π we will use. We will prove the Lemma in Section 4.2.2 next subsubsection.

Lemma 4.3 (Randomized Combinatorial Lemma) *Let A be logspace k-truth-table reducible to a set S by a randomized two-sided error reduction for some $k \in \mathbb{N}$, and let $D : \Sigma^* \to 2^{\Sigma^*}$ be a set system. Suppose that the reduction queries strings of length at most $s(n)$, uses $r(n)$ random bits, and has confidence at least $\delta(n)$ on inputs of $D(y)$ for $y \in \Sigma^n$. Then there are Boolean functions σ and π of 2 variables such that for any $y \in \Sigma^n$,*

$$\Pr_{z \in_u D(y), |\rho| = r(n)}[\sigma(z, \rho)] \geqslant \frac{\delta(n)^2}{2(c_S(s(n)) + 1)^{2k}} \tag{16}$$

$$\Pr_{z \in_u D(y), |\rho| = r(n)}[\pi(z, \rho) = \chi_A(z) \mid \sigma(z, \rho)] \geqslant \frac{1}{2} + \frac{\delta(n)}{4}. \tag{17}$$

Moreover, if we can uniformly sample $D(y)$ in randomized space $O(\log n)$, and $E : \Sigma^* \to 2^{\Sigma^* \times \Sigma^*}$ is a set system such that $E(y) \subseteq D(y) \times \Sigma^{r(|y|)}$ and E can be generated in randomized logspace (given two-way read access to a random bit tape), then for any polynomial $p(n)$, on input $y \in \Sigma^n$ we can generate a collection of lists at least one of which equals $((\sigma(z, \rho), \pi(z, \rho))_{(z,\rho) \in E(y)}$ with confidence at least $1 - \exp(\frac{-\delta(n)^2 p(n)}{(c_S(s(n))+1)^{2k}})$, using logarithmic work space and two-way read access to a random bit tape.

We can apply this lemma to the set A_5, since A_5 belongs to P, whence by hypothesis logspace k-truth-table reduces to a sparse set S for some $k \in \mathbb{N}$. We choose $D(y) \doteq \{\langle y, w \rangle | \, w \in (\mathrm{GF}(2))^m\}$, where $y \doteq \langle C, x \rangle$ and m is the number of gates of C. Since the reduction is computable in randomized logspace, and inputs of $D(y)$ have length $O(n)$ for $y \in \Sigma^n$, the reduction queries strings of length at most a polynomial $s(n)$, uses a polynomial $r(n)$ number of random bits, and has confidence $\delta(n)$ at least inversely polynomial on inputs from $D(y)$ for $y \in \Sigma^n$. Therefore, (16) and (17) guarantee that $\Pr[\sigma]$ and $\Pr[\pi = \chi_{A_5} \mid \sigma] - \frac{1}{2}$ are at least inversely polynomial in n, and hence also in m. By picking w of length m from the random bit tape, we can uniformly sample $D(y)$ in logspace.

The set $E(y)$ consists of $(2^{\ell(m)} - 1)m$ pairs of the form $(\langle y, w \rangle, \rho_{\langle y,w \rangle})$, where w ranges over the oracle queries the circuit C_m on input 1^m makes, and $\rho_{\langle y,w \rangle}$ is chosen uniformly at random from $\Sigma^{r(n)}$. Since all queries of C_m are made in parallel, the set $E(y) \subseteq D(y) \times \Sigma^{r(n)}$ is well-defined, and it is clear that we can generate the set system E in logspace, given two-way read access to a random bit tape. We can choose the polynomial $p(n)$ to be at least $(c_S(s(n)) + 1)^{2k} \cdot \delta^{-2}(n)$, so by Lemma 4.3 we can generate a collection of lists at least one of which equals $((\sigma(z, \rho), \pi(z, \rho))_{(z,\rho) \in E(y)}$ with probability at least $1 - e^{-1}$, using logarithmic work space and two-way read access to a random bit tape.

For each of these lists, we will simulate the circuit C_m on input 1^m using σ and π as given by the list to answer the oracle queries to $B(y, \cdot)$. With probability no less than $\frac{1}{2} - e^{-1}$, at least one of these simulations will produce a list of vectors that contains g. The same gate assignment checker as in the Main Theorem allows to weed out all other vectors.

It is straightforward to check that the resulting algorithm for CVP is in R*L. □

Except for the reduction, we can actually implement every component of our algorithm for CVP in RNC^1. Since all queries to the reduction are asked in parallel, and CVP is complete for P under NC^1-computable many-one reductions, we also obtain the following:

Theorem 4.4 *There is no sparse hard set for* P *under randomized two-sided error bounded truth-table reductions computable in* RNC^1 *and with confidence at least inversely polynomial unless* $\mathrm{P} \subseteq \mathrm{RNC}^1$.

4.2.2 Proof of the Randomized Combinatorial Lemma

Suppose A logspace k-truth-table reduces to a set S for some $k \in \mathbb{N}$ under a randomized two-sided error reduction, and $D : \Sigma^* \to 2^{\Sigma^*}$ is a set system such that the reduction queries strings of length at most $s(n)$, uses $r(n)$ random bits, and has confidence at least $\delta(n)$ on inputs of $D(y)$ for $y \in \Sigma^n$. For a given $y \in \Sigma^n$, we would like to find a subset of $D(y) \times \Sigma^{r(n)}$ of high measure on which the success probability of the randomized reduction is almost as good as the overall success probability, and such that we can guess efficiently which instances of a given list belong to the subset (predicate σ), and for these determine the outcome of the reduction (predicate π). As in the Combinatorial Lemma, we will guarantee the latter condition by making sure that for the subset constructed we know the membership to S of the queries the reduction makes. This will again involve finding a set P of jointly popular queries in S, and a collection of intervals disjoint from S, and the subset will consist of those instances for which the reduction queries all of P and the other queries are in the union of the intervals (and hit each of the intervals). The counting is a bit different from the deterministic case, and goes as follows:

Proof (of the Randomized Combinatorial Lemma)
Let $y \in \Sigma^n$. For $z \in D(y)$ and $\rho \in \Sigma^{r(n)}$, let $Q(z,\rho)$ denote the queries the reduction makes on input z and random seed ρ. D will be short for $D(y)$, and $Q(D)$ will denote the set of all queries the reduction makes on inputs from D and random seeds of length $r(n)$. We will write $e^P(z,\rho)$ for the Boolean indicating whether the reduction accepts the input z on random seed ρ using P instead of S to determine the membership of the queries to S.

We first construct the Boolean functions σ and π.

Claim 4.1 *There exists a set $P \subseteq S$ such that*

$$\begin{aligned}\Pr_{z \in_u D, |\rho| = r(n)}[Q(z,\rho) \cap S = P] &\geqslant \frac{\delta(n)}{(c_S(s(n))+1)^k} \\ \Pr_{z \in_u D, |\rho| = r(n)}[e^P(z,\rho) = \chi_A(z) \mid Q(z,\rho) \cap S = P] &\geqslant \frac{1}{2} + \frac{\delta(n)}{2}.\end{aligned}$$

This is because if there were no such set P, then

$$\begin{aligned}\Pr[e^S(z,\rho) = \chi_A(z)] - \frac{1}{2} &= \sum_{P \subseteq S \cap \Sigma^{\leqslant s(n)}} \Pr[P] \cdot (\Pr[e^S(z,\rho) = \chi_A(z) \mid P] - \frac{1}{2}) \\ &= \sum_{\substack{P \subseteq S \cap \Sigma^{\leqslant s(n)} \\ \Pr[P] \geqslant \frac{\delta(n)}{(c_S(s(n))+1)^k}}} \Pr[P] \cdot (\Pr[e^S(z,\rho) = \chi_A(z) \mid P] - \frac{1}{2}) + \\ &\quad \sum_{\substack{P \subseteq S \cap \Sigma^{\leqslant s(n)} \\ \Pr[P] < \frac{\delta(n)}{(c_S(s(n))+1)^k}}} \Pr[P] \cdot (\Pr[e^S(z,\rho) = \chi_A(z) \mid P] - \frac{1}{2}) \\ &< \frac{\delta(n)}{2} + \sum_{i=0}^{k} \binom{c_S(s(n))}{i} \cdot \frac{\delta(n)}{(c_S(s(n))+1)^k} \cdot \frac{1}{2} \\ &\leqslant \delta(n),\end{aligned}$$

where we use P as a shortcut for the event that $Q(z,\rho) \cap S = P$, and all probabilities are with respect to the uniform distribution of (z,ρ) over $D \times \Sigma^{r(n)}$. In the last step, we are using the inequality $\sum_{i=0}^{k} \binom{c}{i} \leqslant (c+1)^k$. We obtain a contradiction to the fact that the reduction has confidence $\delta(n)$, and therefore prove Claim 1.

Claim 4.2 *There exists a collection of at most $k - |P|$ intervals I_i, $i \in \mathcal{I}$, of $Q(D) \setminus S$ such that*

$$\Pr_{z \in_u D, |\rho| = r(n)} [Q(z,\rho) \setminus P \subseteq \cup_{i \in \mathcal{I}} I_i \textit{ and } Q(z,\rho) \textit{ hits every } I_i \mid P \subseteq Q(z,\rho)] \geqslant \frac{\delta(n)}{2(c_S(s(n))+1)^{k-|P|}}$$

$$\Pr_{z \in_u D, |\rho| = r(n)} [e^S(z,\rho) = \chi_A(z) \mid P \subseteq Q(z,\rho) \textit{ and } Q(z,\rho) \setminus P \subseteq \cup_{i \in \mathcal{I}} I_i \textit{ and } Q(z,\rho) \textit{ hits every } I_i] \geqslant \frac{1}{2} + \frac{\delta(n)}{4}.$$

This follows from Claim 1 by essentially the same argument. Without loss of generality, we can assume that the reduction always makes exactly k queries. In case $|P| = k$ or $c_S(s(n)) = 0$, Claim 1 immediately implies Claim 2. Assume $|P| < k$ and $c_S(s(n)) > 0$. The set S divides $Q(D) \setminus S$ into the union of m intervals $I_1, \ldots, I_m$ ($1 \leqslant m \leqslant c_S(s(n))$), and the reduction maps each instance to a collection of 1 up to $k - |P|$ of these intervals. Suppose there were no subset $\mathcal{I}$ of $\{1, \ldots, m\}$ such that $\Pr[\mathcal{I} \mid P] \geqslant \frac{\delta(n)}{2(c_S(s(n))+1)^{k-|P|}}$ and $\Pr[e^S(z,\rho) = \chi_A(z) \mid \mathcal{I} \text{ and } P] \geqslant \frac{1}{2} + \frac{\delta(n)}{4}$, where we use P as a shortcut for the event that $Q(z,\rho) \cap S = P$, $\mathcal{I}$ for the event that $Q(z,\rho) \setminus P \subseteq \cup_{i \in \mathcal{I}} I_i$ and $Q(z,\rho)$ hits every I_i for $i \in \mathcal{I}$, and again all probabilities are with respect to the uniform distribution of (z,ρ) over $D \times \Sigma^{r(n)}$. Then

$$\begin{aligned}
&\Pr[e^S(z,\rho) = \chi_A(z) \mid Q(z,\rho) \cap S = P] - \frac{1}{2} \\
&\quad = \sum_{\varnothing \neq \mathcal{I} \subseteq \{1,\ldots,m\}} \Pr[\mathcal{I} \mid P] \cdot (\Pr[e^S(z,\rho) = \chi_A(z) \mid \mathcal{I} \text{ and } P] - \frac{1}{2}) \\
&\quad = \sum_{\substack{\varnothing \neq \mathcal{I} \subseteq \{1,\ldots,m\} \\ \Pr[\mathcal{I} \mid P] \geqslant \frac{\delta(n)}{2(c_S(s(n))+1)^{k-|P|}}}} \Pr[\mathcal{I} \mid P] \cdot (\Pr[e^S(z,\rho) = \chi_A(z) \mid \mathcal{I} \text{ and } P] - \frac{1}{2}) + \\
&\qquad \sum_{\substack{\varnothing \neq \mathcal{I} \subseteq \{1,\ldots,m\} \\ \Pr[\mathcal{I} \mid P] < \frac{\delta(n)}{2(c_S(s(n))+1)^{k-|P|}}}} \Pr[\mathcal{I} \mid P] \cdot (\Pr[e^S(z,\rho) = \chi_A(z) \mid \mathcal{I} \text{ and } P] - \frac{1}{2}) \\
&\quad < \frac{\delta(n)}{4} + \sum_{i=1}^{k-|P|} \binom{m}{i} \cdot \frac{\delta(n)}{2(c_S(s(n))+1)^{k-|P|}} \cdot \frac{1}{2} \\
&\quad \leqslant \frac{\delta(n)}{2},
\end{aligned}$$

which contradicts Claim 1, and that way proves Claim 2. In the last step, we used the inequality $\sum_{i=0}^{\ell} \binom{m}{i} \leqslant (m+1)^{\ell}$ again.

We define $\sigma(z,\rho)$ to be the Boolean indicating whether $P \subseteq Q(z,\rho)$ and $Q(z,\rho) \setminus P \subseteq \cup_{i \in \mathcal{I}} I_i$ and $Q(z,\rho)$ hits every I_i for $i \in \mathcal{I}$, and $\pi(z,\rho)$ to be $e^P(z,\rho)$. Claims 1 and 2 combined show that σ and π satisfy conditions (16) and (17).

Now, we will see how we can produce with confidence at least $1-\exp(\frac{-\delta(n)^2 p(n)}{(c_S(s(n))+1)^{2k}})$ a collection of lists at least one of which equals $((\sigma(z,\rho),\pi(z,\rho))_{(z,\rho)\in E(y)}$, using logarithmic work space and two-way read access to a random bit tape.

We generate a multiset F of $2p(n)$ instances chosen uniformly at random from $D \times \Sigma^{r(n)}$. With probability at least $1-(1-\frac{\delta(n)^2}{2(c_S(s(n))+1)^{2k}})^{2p(n)} \geqslant 1-\exp(\frac{-\delta(n)^2 p(n)}{(c_S(s(n))+1)^{2k}})$, F contains an instance satisfying σ. Then, as in the Combinatorial Lemma, we will consider, for every instance of F, every subset of the queries the reduction makes on that instance as a candidate P' for P, and consider every collection of k or fewer intervals of the set of all queries on inputs from $E(y)$ as candidates for the intervals I_i, $i \in \mathcal{I}$. For each of these candidates, we output the list consisting of a pair for each instance of $E(y)$, where the pair corresponding to (z,ρ) consists of

- a Boolean indicating whether $P' \subseteq Q(z,\rho)$, $Q(z,\rho)\setminus P$ is in the union of the candidate intervals, and each of the candidate intervals is hit by $Q(z,\rho)$, and
- the Boolean $e^{P'}(z,\rho)$, indicating whether the reduction accepts z on random seed ρ using P' to decide membership to S of the queries.

If F contains at least one instance satisfying σ, $((\sigma(z,\rho),\pi(z,\rho))_{(z,\rho)\in E(y)}$ will be among the lists produced. Moreover, by the argument given in the proof of the Combinatorial Lemma, it is clear that we can generate these lists in randomized logspace, provided two-way read access to the random bit tape. We need the two-way read access to regenerate the instances of $E(y)$ and F when needed. □

Note that applying the Randomized Combinatorial Lemma to deterministic reductions yields a result similar to the Combinatorial Lemma, but the lower bound corresponding to equation (10) is weaker: the fraction of $A \cap D(y)$ we obtain here is essentially the square of the fraction the Combinatorial Lemma guarantees.

4.3 Randomized Generic Theorem for P

In this section, we generalize the Randomized Main Theorem to a Randomized Generic Theorem for P by parameterizing the sparseness condition, and the space bound, the bound on the number of truth-table queries and the confidence of the randomized reduction. We also consider some instantiations of the Randomized Generic Theorem other than the Randomized Main Theorem, analogous to the ones made in the deterministic case.

4.3.1 Statement

The Randomized Combinatorial Lemma parameterizes to:

Lemma 4.5 (Randomized Generic Combinatorial Lemma) *Let A be truth-table reducible to a set S by a randomized two-sided error reduction, and let $D : \Sigma^* \to 2^{\Sigma^*}$ be a set system. Suppose that the reduction queries at most $k(n)$ strings of length at most $s(n)$, uses $r(n)$ random bits and work space at most $a(n)$ (given two-way read access to a random bit tape), and has confidence at least $\delta(n)$ on inputs of $D(y)$ for $y \in \Sigma^n$. Then there are Boolean functions σ and π of 2 variables* *such that for any $y \in \Sigma^n$,*

$$\begin{aligned}
\Pr_{z \in_u D(y), |\rho| = r(n)} [\sigma(z,\rho)] &\geqslant \frac{\delta(n)^2}{2(c_S(s(n))+1)^{2k}} \\
\Pr_{z \in_u D(y), |\rho| = r(n)} [\pi(z,\rho) = \chi_A(z) \mid \sigma(z,\rho)] &\geqslant \frac{1}{2} + \frac{\delta(n)}{4}.
\end{aligned}$$

Moreover, suppose we can uniformly sample $D(y)$ in randomized space $b(n)$, and $E : \Sigma^* \to 2^{\Sigma^* \times \Sigma^*}$ is a set system such that $E(y) \subseteq D(y) \times \Sigma^{r(|y|)}$ and we can generate E using at most $b(n)$ work space (given two-way read access to a random bit tape). Then for any function $p : \mathbb{N} \to \mathbb{N}$ such that $\log p(n)$ is space constructible, on input $y \in \Sigma^n$, we can generate a collection of lists at least one of which equals $((\sigma(z,\rho), \pi(z,\rho))_{(z,\rho) \in E(y)}$ with confidence at least $1 - \exp(\frac{-\delta(n)^2 p(n)}{(c_S(s(n))+1)^{2k}})$, using work space $O(k(n) \cdot (a(n) + b(n)) + \log p(n) + \log r(n))$ and two-way read access to a random bit tape, provided $\log r(n)$ is space constructible.

Proof

The proof is a straightforward combination of the one for the Randomized Combinatorial Lemma and the space bound analysis given in the proof of the Generic Combinatorial Lemma. □

Plugging in this lemma in the proof of the Randomized Main Theorem yields:

Theorem 4.6 (Randomized Generic Theorem) *There is no hard set for* P *of census at most $c(n)$ under randomized two-sided error truth-table reductions that query at most $O(k(n))$ strings of length at most $O(s(n))$, run in space $O(a(n))$ (given two-way read access to a random bit tape) and have confidence at least $\Omega(\delta(n))$ on inputs of length n unless* $\mathrm{P} \subseteq \mathrm{ZP^*SPACE}[t(n^{O(1)})]$, *where*

$$\begin{aligned}
t(n) &= k(n) \cdot (a(n) + b(n)) \\
b(n) &= \log n + k(n) \cdot \log c(s(n)) + \log \delta^{-1}(n),
\end{aligned}$$

provided the following technical conditions hold: $b(n)$ is space constructible, $a(O(n)) \subseteq O(a(n))$, $\log c(O(s(O(n)))) \subseteq O(\log c(s(n)))$, $k(O(n)) \subseteq O(k(n))$, and $\log \delta^{-1}(O(n)) \subseteq O(\log \delta^{-1}(n))$.

Proof

Since CVP is complete for P under logspace many-one reductions and P is closed under complement, it suffices to show that we can solve CVP in $\mathrm{ZP^*SPACE}[t(n)]$ assuming the existence of a set S of census at most $c(n)$ that is hard for P under randomized two-sided error truth-table reductions that query at most $O(k(n))$ strings of length at most $O(s(n))$, run in space $O(a(n))$ (provided two-way read access to a random bit tape) and have confidence at least $\Omega(\delta(n))$ on inputs of length n.

The algorithm is the same as in the proof of the Randomized Main Theorem. We only have to analyze its space complexity under the parameterizations considered.

Let y be an instance of CVP of length n with m gates. Under the given technical conditions, the space needed for the simulations of the randomized circuit C_m of Lemma 4.2 using an oracle is $O(\ell(m)) \subseteq O(b(n))$. Consequently, the space complexity of the set system E is $O(b(n) + \log r(n)) \subseteq O(b(n) + a(n))$. In order for the success probability to be bounded from below by a constant, we choose $\log p(n) \in \theta(b(n))$ in the application of the Randomized Generic Combinatorial Lemma. The space needed to construct the lists with oracle answers is then $O(t(n))$. Since the simulations themselves only require $O(b(n))$ space as argued above, and the solution checker runs in logspace, the overall space complexity of the algorithm is $O(t(n))$. □

4.3.2 Instantiations

By the same simple observations as in the deterministic case, we obtain the following instantiations of the Randomized Generic Theorem for polynomially dense hard sets. The Randomized Main Theorem is a special case of the first one.

Theorem 4.7 *Let $e \geqslant 1$ and $f \geqslant 0$. There is no sparse hard set for* P *under randomized two-sided error truth-table reductions computable in space $O(log^e n)$ (given two-way read access to a random bit tape) with confidence at least inversely polynomial that make at most $O(\log^f n)$ queries unless* $\mathrm{P} \subseteq \mathrm{ZP^*SPACE}[\log^{e+2f} n]$.

Theorem 4.8 *Let $e \geqslant 1$, $f \geqslant 0$ and $g > 0$. There is no sparse hard set for* P *under randomized two-sided error truth-table reductions computable in space $O(log^e n)$ (given two-way read access to a random bit tape) with confidence at least $2^{-O(\log^g n)}$ that make at most $O(\log^f n)$ queries unless* $\mathrm{P} \subseteq \mathrm{ZP^*SPACE}[\log^a n]$, *where $a = \max(e + 2f, f + g)$.*

For quasipolynomially dense hard sets we get:

Theorem 4.9 *Let $d > 0$, $e \geqslant 1$ and $f, g \geqslant 0$. There is no hard set for* P *with density bounded by $2^{O(\log^d n)}$ under randomized two-sided error truth-table reductions computable in space $O(\log^e n)$ (given two-way read access to a random bit tape) with confidence at least $2^{-O(\log^g n)}$ that make at most $O(\log^f n)$ queries unless* $\mathrm{P} \subseteq \mathrm{ZP^*SPACE}[\log^a n]$, *where $a = \max(de + 2f, e + f, f + g)$.*

In particular the following holds:

Corollary 4.10 *There is no quasipolynomially dense hard set for* P *under randomized two-sided error truth-table reductions computable in polylog-space (given two-way read access to a random bit tape) with confidence at least inversely quasipolynomial and using no more than polylogarithmically many queries unless* $\mathrm{P} \subseteq \mathrm{ZP^*SPACE}[\mathrm{polylog} n]$.

4.4 Extension to classes other than P

In this subsection, we apply the idea of the Randomized Main Theorem to prove theorems of the form: There is no sparse hard set for $\mathcal{C}_1$ under randomized two-sided error bounded truth-table reductions with confidence at least inversely polynomial and computable within the randomized class $\mathcal{C}_2$ unless $\mathcal{C}_1 \subseteq \mathcal{C}_2$. For the technique to work, sets in $\mathcal{C}_1$ must have unique membership proofs that can be constructed in $\mathcal{C}_1$ and checked in $\mathcal{C}_2$, and $\mathcal{C}_2$ has to contain RNC^1. We can then Hadamard encode the membership proofs and use the Randomized Combinatorial Lemma and Lemma 4.2 to recover them with high probability in $\mathcal{C}_2$. As in the deterministic case, this approach works for NL and L. In the randomized case, we can also apply it to NP. Sets in NP are not known to have unique membership proofs, but Valiant and Vazirani [32] showed that randomness allows us to reduce the number of proofs with high probability to one, if there is one.

Theorem 4.11 *There is no sparse hard set for* NL *under randomized two-sided error bounded truth-table reductions computable in logspace (given two-way read access to a random bit tape) and with confidence at least inversely polynomial unless* $\mathrm{NL} \subseteq \mathrm{ZP^*L}$.

Proof
Assuming the existence of a sparse hard set for NL under randomized two-sided error bounded truth-table reductions computable in logspace (given two-way read access to a random bit tape) and with confidence at least inversely polynomial, we will show how to solve in $\mathrm{R^*L}$ the NL-complete problem DAG-STCON. Since NL is closed under complement [17, 31], this suffices to show that $\mathrm{NL} \subseteq \mathrm{ZP^*L}$.

We use the auxiliary set

$$A_6 = \{\langle G, s, t, w\rangle \,|\, G \text{ DAG with } m \text{ vertices, } s, t \text{ vertices of } G,\ w \in (\mathrm{GF}(2))^m \text{ and } \sum_{j=1}^{m} w_j g_j = 1\},$$

where g_j is a Boolean indicating whether G contains a directed path from the j-th vertex of G to t, and the arithmetic is over GF(2) . We argued in the proof of Theorem 3.12 that the values $g_1, \ldots, g_m$ constitute a logspace verifiable proof of membership of $\langle G, s, t\rangle$ to DAG-STCON.

A_6 incorporates Hadamard encodings of these values. Since the set of DAGs is in coNL = NL, and we can compute the values g_j and check the equation $\sum_{j=1}^{m} w_j g_j = 1$ in $\mathrm{L^{NL}} = \mathrm{NL}$, it is obvious that $A_6 \in \mathrm{NL}$, whence by hypothesis bounded truth-table reduces to a sparse set through a randomized two-sided error reduction computable in logspace (provided two-way read access to a random bit tape) and with confidence at least inversely polynomial. Therefore, applying Lemma 4.2 and the Randomized Combinatorial Lemma as in the proof of the Randomized Main Theorem, allows to generate in $\mathrm{R^*L}$ a list of strings in Σ^m which contains $g_1 g_2 \ldots g_m$ with probability bounded from 0. So, it suffices to submit each of these strings to the logspace proof checker to obtain our $\mathrm{R^*L}$ algorithm for DAG-STCON. □

The above algorithm for DAG-STCON is actually $\mathrm{RNC^1}$ modulo the complexity of the reduction and has only parallel queries to the reduction. DAG-STCON being complete for NL under $\mathrm{NC^1}$-computable many-one reductions, it follows that:

Theorem 4.12 *There is no sparse hard set for* NL *under randomized two-sided error bounded truth-table reductions computable in* $\mathrm{RNC^1}$ *and with confidence at least inversely polynomial unless* $\mathrm{NL} \subseteq \mathrm{RNC^1}$.

For L we obtain:

Theorem 4.13 *There is no sparse hard set for* L *under randomized two-sided error bounded truth-table reductions computable in* $\mathrm{RNC^1}$ *and with confidence at least inversely polynomial unless* $\mathrm{L} \subseteq \mathrm{RNC^1}$.

Proof
Assuming the existence of a sparse hard set for L under randomized two-sided error bounded truth-table reductions computable in $\mathrm{RNC^1}$ and with confidence at least inversely polynomial, we will show how to solve in $\mathrm{RNC^1}$ the L-complete problem F-STCON.

Consider the auxiliary set

$$A_7 = \{\langle G, s, t, w\rangle \,|\, G \text{ forest with } m \text{ edges, } s, t \text{ vertices of } G,\ w \in (\mathrm{GF}(2))^m \text{ and } \sum_{j=1}^{m} w_j e_j = 1\},$$

where e_j is a Boolean indicating whether s and t belong to the same component of G and the j-th edge of G is on the unique path connecting s and t in G. The arithmetic is over GF(2) . We showed in the proof of Theorem 3.14 that the values $e_1, \ldots, e_m$ constitute an NC^1-verifiable proof of membership of $\langle G, s, t\rangle$ to F-STCON. It is also obvious from the arguments in that proof that $A_7 \in \mathrm{L}$.

A_7 contains Hadamard encodings of the values $e_1, \ldots, e_m$, and by hypothesis, it bounded truth-table reduces to a sparse set through a randomized two-sided error reduction computable in RNC^1 and with confidence at least inversely polynomial. Therefore, another application of Lemma 4.2 and the Randomized Combinatorial Lemma as in the proof of the Randomized Main Theorem, yields an RNC^1 algorithm that with probability bounded from 0, outputs a list containing the string $e_1 e_2 \ldots e_m$. We can then use the NC^1 solution checker to weed out the incorrect ones. The resulting algorithm for F-STCON is in RNC^1. $\square$

Valiant and Vazirani [32] constructed a randomized one-sided error many-one reduction of satisfiability to unique satisfiability computable in RNC^1. It allows us to apply the strategy behind the Randomized Main Theorem to NP.

Theorem 4.14 *There is no sparse hard set for* NP *under randomized two-sided error bounded truth-table reductions computable in logspace (given two-way read access to a random bit tape) and with confidence at least inversely polynomial unless* $\mathrm{NP} = \mathrm{R}^*\mathrm{L}$.

Proof
Assuming the existence of a sparse hard set for NP under randomized two-sided error bounded truth-table reductions computable in logspace (given two-way read access to a random bit tape) and with confidence at least inversely polynomial, we will show how to check in $\mathrm{R}^*\mathrm{L}$ the satisfiability of Boolean formulae with at most one satisfying assignment (USAT). This suffices to show that $\mathrm{NP} \subseteq \mathrm{R}^*\mathrm{L}$, because SAT is NP-complete under logspace many-one reductions, and Valiant and Vazirani proved that we can many-one self-reduce SAT in randomized logspace with one-sided error such that with confidence at least inversely polynomial, a satisfiable formula is reduced to one with exactly one satisfying assignment.

Consider the language

$$A_8 = \{\langle \varphi, w\rangle \,|\, \varphi \text{ Boolean formula on } m \text{ variables, } w \in \Sigma^m \text{ and } \exists b \in \Sigma^m : [\varphi(b) \text{ and } w \cdot b = 1]\},$$

where $w \cdot b$ represents the inner product of w and b as vectors in $(\mathrm{GF}(2))^m$. It is clear that $A_8 \in \mathrm{NP}$, whence by hypothesis bounded truth-table reduces to a sparse set under randomized two-sided error reductions computable in logspace (given two-way read access to a random bit tape).

In case φ has exactly one satisfying assignment b^*, $(\chi_{A_8}(\langle \varphi, w\rangle))_{w \in (\mathrm{GF}(2))^m}$ is the Hadamard encoding of this unique satisfying assignment. Using Lemma 4.2 and the Randomized Combinatorial Lemma as in the proof of the Randomized Main Theorem, we can recover b^* in $\mathrm{R}^*\mathrm{L}$ in the sense of generating a list of assignments that contains b^* with bounded positive probability. Since we can check whether a given assignment satisfies φ in logspace, doing that for all assignments of the above list yields an $\mathrm{R}^*\mathrm{L}$ algorithm for the promise problem USAT. $\square$

Again, the algorithm we constructed for SAT is actually RNC^1 modulo the complexity of the reduction, and only has parallel queries to the reduction. Since SAT is complete for NP under NC^1-computable many-one reductions, we obtain:

Theorem 4.15 *There is no sparse hard set for* NP *under randomized two-sided error bounded truth-table reductions computable in* RNC^1 *and with confidence at least inversely polynomial unless* $\mathrm{NP} = \mathrm{RNC}^1$.

Finally, we can also prove an analogue result for (presumably) stronger reductions, namely polynomial-time computable ones.

Theorem 4.16 *There is no sparse hard set for* NP *under randomized two-sided error polynomial-time bounded truth-table reductions with confidence at least inversely polynomial unless* NP = RP.

Proof
The proof follows the lines of the one for Theorem 4.14. It suffices to observe that the Randomized Combinatorial Lemma also holds when the set A is *polynomial-time* bounded truth-table reducible to the sparse set S by a randomized two-sided error reduction, and that the algorithm it provides then also runs in polynomial-time, as does the algorithm for SAT obtained by combination with Lemma 4.2 like in Theorem 4.14. □

5 Open Problems

It is easy to see that our Generic Theorem, based on the hypothesis that P $\neq$ L, only rules out the existence of sparse hard sets for P under logspace truth-table reductions with a *bounded* number of queries, even if we relax the sparseness condition. If we allow the number of queries to increase modestly with the input size, e.g., polylogarithmically, we get other unlikely inclusions of P in space bounded complexity classes. However, without any explicit bound on the number of queries, we do not get a collapse at all. Can we do better using another technique?

Regarding randomized reductions with two-sided error, it is an open question whether our results also hold for the more natural read-once concept for randomized space bounded computation. In this model, the Turing machine only has one-way read access to its random tape, as opposed to the multiple access randomness model, in which the machine has two-way read access to its random tape. The randomized algorithms we construct use the latter model, because they basically simulate randomized circuits, and randomized circuits inherently have the ability to reuse their random seeds. Nisan [25] gives strong evidence that in general the multiple access model is more powerful than the read-once model, but in this specific context that may not be the case.

Acknowledgments

I would like to thank Ashish Naik for explaining the results of Cai, Naik and Sivakumar [7], and both Lance Fortnow and Ashish Naik for interesting discussions, comments, and for proofreading the paper. I also thank Jin-Yi Cai, Mitsu Ogihara, and D. Sivakumar for their encouragement, and two anonymous referees for their helpful comments.

References

[1] N. Alon, O. Goldreich, J. Håstad, and R. Peralta. Simple constructions of almost k-wise independent random variables. In *Proceedings of the 31st IEEE Symposium on Foundations of Computer Science*, pages 544–553, New York, 1990. IEEE.

[2] V. Arvind, J. Köbler, and M. Mundhenk. On bounded truth-table, conjunctive, and randomized reductions to sparse sets. In *Proceedings of the 12th Conference on Foundations of Software*

Technology and Theoretical Computer Science, volume 652 of *Lecture Notes in Computer Science*, pages 140–151, Berlin, 1992. Springer-Verlag.

[3] J. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I*, volume 11 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1995.

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

[5] A. Borodin, J. von zur Gathen, and J. Hopcroft. Fast parallel matrix and GCD computations. *Information and Control*, 52:241–256, 1982.

[6] J. Cai, A. Naik, and D. Sivakumar. Bounded truth-table reductions of P. Technical Report TR-95-42, Department of Computer Science, State University of New York at Buffalo, September 1995.

[7] J. Cai, A. Naik, and D. Sivakumar. On the existence of hard sparse sets under weak reductions. In *Proceedings of the 13th Symposium on Theoretical Aspects of Computer Science*, volume 1046 of *Lecture Notes in Computer Science*, pages 307–318, Berlin, 1996. Springer-Verlag.

[8] J. Cai and D. Sivakumar. The resolution of a Hartmanis conjecture. In *Proceedings of the 36th IEEE Symposium on Foundations of Computer Science*, pages 362–371, New York, 1995. IEEE.

[9] J. Cai and D. Sivakumar. Resolution of Hartmanis' conjecture for NL-hard sparse sets. Technical Report TR-95-40, Department of Computer Science, State University of New York at Buffalo, September 1995.

[10] S. Cook. A taxonomy of problems with fast parallel algorithms. *Information and Control*, 64:2–22, 1985.

[11] S. Cook and P. McKenzie. Problems complete for deterministic logarithmic space. *Journal of Algorithms*, 8:385–394, 1987.

[12] W. Eberly. Very fast parallel polynomial arithmetic. *SIAM Journal on Computing*, 18(5):955–976, 1989.

[13] O. Goldreich and L. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the 21st ACM Symposium on the Theory of Computing*, pages 25–32, New York, 1989. ACM.

[14] J. Hartmanis. On log-tape isomorphisms of complete sets. *Theoretical Computer Science*, 7(3):273–286, 1978.

[15] S. Homer and L. Longpré. On reductions of NP sets to sparse sets. *Journal of Computer and System Sciences*, 48(2):324–336, 1994.

[16] J. Hong. On some deterministic space complexity problems. In *Proceedings of the 12th ACM Symposium on the Theory of Computing*, pages 310–317, New York, 1980. ACM.

[17] N. Immerman. Nondeterministic space is closed under complementation. *SIAM Journal on Computing*, 17(5):935–938, 1988.

[18] R. Karp and R. Lipton. Turing machines that take advice. *L'Enseignement Mathématique*, 28(2):191–209, 1982. A preliminary version appeared in STOC 1980.

[19] R. Ladner. The circuit value problem is log-space complete for P. *SIGACT News*, 6(2):18–20, 1975.

[20] R. Ladner and N. Lynch. Relativization of questions about logspace computability. *Mathematical Systems Theory*, 10(1):19–32, 1976.

[21] F. MacWilliams and N. Sloane. *The Theory of Error-Correcting Codes*. North-Holland, 1981.

[22] 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.

[23] K. Mulmuley. A fast parallel algorithm to compute the rank of a matrix over an arbitrary field. *Combinatorica*, 7(1):101–104, 1987.

[24] J. Naor and M. Naor. Small-bias probability spaces: efficient constructions and applications. In *Proceedings of the 22nd ACM Symposium on the Theory of Computing*, pages 213–223, New York, 1990. ACM.

[25] N. Nisan. On read-once vs. multiple access to randomness in logspace. *Theoretical Computer Science*, 107:135–144, 1993.

[26] M. Ogihara. Sparse P-hard sets yield space-efficient algorithms. In *Proceedings of the 36th IEEE Symposium on Foundations of Computer Science*, pages 354–361, New York, 1995. IEEE.

[27] M. Ogiwara and A. Lozano. On one-query self-reducible sets. In *Proceedings of the 6th IEEE Structure in Complexity Theory Conference*, pages 139–151, New York, 1991. IEEE.

[28] M. Ogiwara 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.

[29] D. Ranjan and P. Rohatgi. On randomized reductions to sparse sets. In *Proceedings of the 7th IEEE Structure in Complexity Theory Conference*, pages 239–242, New York, 1992. IEEE.

[30] W. Savitch. Maze recognizing automata and nondeterministic tape complexity. *Journal of Computer and System Sciences*, 7:389–403, 1973.

[31] R. Szelepcsényi. The method of forced enumeration for nondeterministic automata. *Acta Informatica*, 26:279–284, 1988.

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

[33] J. H. van Lint. *Introduction to Coding Theory*. Springer-Verlag, 1991.

[34] C. Wilson. Relativized NC. *Mathematical Systems Theory*, 20:13–29, 1987.

Dieter van Melkebeek
Department of Computer Science
The University of Chicago
1100 East 58th Street
Chicago, IL 60637
USA
Email: dieter@cs.uchicago.edu
WWW: http://www.cs.uchicago.edu/~dieter