**Open Problems Column**
**Edited by William Gasarch**

This issue's Open Problem Column is by Ryan Williams and is on *Some Open Problems Regarding Lower Bounds For NP*. Ryan himself has made progress on lower bounds for SAT and is an expert in the area.

I invite any reader who has knowledge of some area to contact me and arrange to write a column about open problems in that area. That area can be (1) broad or narrow or anywhere inbetween, and (2) really important or really unimportant or anywhere inbetween.

## Some Open Problems Regarding Lower Bounds For NP

by Ryan Williams

It looks obvious that $\mathsf{LOGSPACE} \neq \mathsf{NP}$, but it also looks like we are far from proving it. In 1997, Fortnow [For00] proved that at least one of the following is true:

- SAT is not solvable in $O(n \cdot \text{poly}(\log n))$ time.
- $\mathsf{NP} \neq \mathsf{LOGSPACE}$.

In complexity class notation, we may write this as

$$\text{SAT} \notin \mathsf{TIME}[n \cdot \text{poly}(\log n)] \cap \mathsf{LOGSPACE}.$$

Fortnow's lower bound holds for general-purpose algorithms, including pointer machines, word RAMs, etc. In that sense, his method truly exploits the nondeterministic expressiveness of the SAT problem, rather than artifacts of a particular machine model. A nice property of such "general purpose" lower bounds is that there is nothing particularly special about SAT: a host of other NP-complete problems such as Vertex Cover, Hamiltonian Path, and Max Cut also exhibit essentially the same lower bound (with some notable exceptions, as we shall see below). However, in the following we stick to SAT for simplicity. (In fact, one may think of "SAT" as "satisfiability of 3-CNF formulas.")

Here we present some open problems concerning possible extensions of this basic lower bound.

**Improve the State of Affairs.** After Fortnow's paper, there was a short flurry of work in the late 90s and early 2000s regarding time-space tradeoff lower bounds for SAT and related problems (surveyed in [vM07]). Note that $\mathsf{LOGSPACE} \neq \mathsf{NP}$ is equivalent to "there is no algorithm for SAT running in $n^k$ time and $k \log n$ space, for *every* constant $k$," so it is a significant open problem to understand the largest $k$ for which such statements can be proved. The state of the art in this direction, as reported in several papers [Wil08, Wil13, BW15], is that there is no algorithm for SAT running in $n^{2\cos(\pi/7)-o(1)}$ time and $n^{o(1)}$ space, simultaneously. In complexity class notation, we write this as

$$\text{SAT} \notin \mathsf{DTISP}[n^{2\cos(\pi/7)-o(1)}, n^{o(1)}].$$

Unfortunately, $2\cos(\pi/7) < 1.802$, so this is not even a quadratic lower bound. Buss and I proved in a rigorous sense that current techniques *cannot* exceed the curious $2\cos(\pi/7)$ exponent, so there is a real barrier here [BW15]. An immediate open problem is:

> **Problem 1:** *Improve the time lower bound for solving SAT on log-space machines. Even showing that there is an $\varepsilon > 0$ such that SAT is not in $\mathsf{DTISP}[n^{2\cos(\pi/7)+\varepsilon}, O(\log n)]$ would be interesting.*

It is clear that some genuinely new ideas are needed to resolve Problem 1. Fortnow's approach (followed by all others) is to give a proof by contradiction: show that a super-fast SAT algorithm would imply a too-good-to-be-true speedup: for example, nondeterministic time $t$ is contained in nondeterministic time $t^{0.999}$ for some time bound $t$, which contradicts the time hierarchy theorem for nondeterminism. Perhaps we could use a hypothetical super-fast SAT algorithm to contradict other known lower bounds (such as circuit/formula lower bounds) instead of those based on a time hierarchy?

**Boolean Formulas.** A special case of log-space computations are LOGTIME-uniform Boolean De Morgan formulas. A *De Morgan formula family of size $s(n)$* is a collection of formulas $\{F_n\}$ such that $F_n$ has $n$ variables, gates of the form AND, OR, or NOT of fan-in two, and at most $s(n)$ leaves. Such a family is *LOGTIME-uniform* if there is an algorithm which can return any desired bit of the encoding of $F_n$ in only $O(\log n)$ time (see Vollmer [Vol99] for a formal definition). Proving that SAT does not have LOGTIME-uniform De Morgan formulas of $n^k$ size, for every constant $k$, is equivalent to proving $\mathsf{NC}^1 \neq \mathsf{NP}$.

Without the LOGTIME-uniform condition, it is known that there are functions computable in linear time (namely, Andreev's function [And87, Hås98]) which require Boolean formulas of size $n^{3-o(1)}$. By tight reductions, this implies that SAT needs formulas of at least $n^{3-o(1)}$ size as well (with a slightly worse $o(1)$ factor). Can we prove a stronger formula-size lower bound assuming that the formulas are LOGTIME-uniform?

**Problem 2:** *Prove that SAT requires LOGTIME-uniform De Morgan formulas of $n^{3+\varepsilon}$ size, for some $\varepsilon > 0$.*

Intuitively, one could hope to combine both the diagonalization and simulation-based techniques of Fortnow with the combinatorial/variable restriction ideas in the lower bounds for Andreev's function. These are somewhat orthogonal approaches to lower bounds that together could potentially be "amplified" to prove something stronger.

**TC0 Circuits.** Depth-three TC0 circuits, composed of MAJORITY gates of unbounded fan-in and NOT gates, represent a very difficult "frontier" of circuit lower bounds. Exponential-gate lower bounds for depth-two circuits have been known for a long time [HMP$^+$93], but for depth-three, even non-linear gate lower bounds were only recently obtained. Recently, Kane and I proved that an extension of Andreev's function requires depth-three circuits composed of MAJORITY and NOT gates with at least $n^{1.5-o(1)}$ gates and $n^{2.5-o(1)}$ wires, via random restriction methods [KW16]. This suggests the problem:

**Problem 3:** *Prove that SAT requires LOGTIME-uniform depth-three MAJORITY/NOT circuits of $n^{2.5+\varepsilon}$ wires, for some $\varepsilon > 0$.*

In earlier work, Allender and Koucky [AK10] proved that for every $d$, there is an $\varepsilon \in (0, 1)$ such that SAT cannot be solved by LOGTIME-uniform depth-$d$ MAJORITY/NOT circuits with $O(n^{1+\varepsilon})$ wires. Their lower bound works by showing that, if such circuits existed, then we'd have a provably false time-space tradeoff algorithm for SAT (appealing to previously proved lower bounds). Can these ideas be combined with the random restriction methods to prove an even stronger lower bound? One could imagine setting some variables of a particularly structured SAT formula at random, in such a way that it does not "kill" the formula entirely, but rather leaves behind "hard" sub-formulas from some class that are still difficult to solve for another (diagonalization-based) reason.

**Max Clique.** Due to tight reductions between many $\mathsf{NP}$-complete problems, any "general purpose" lower bound for SAT extends to other problems such as Vertex Cover, Independent Set, and Max Cut. These problems require $2^{\Omega(n)}$ time to solve (where $n$ is either the number of variables or the number of nodes) assuming the Exponential Time Hypothesis (ETH) [IPZ01], even on $O(n)$-edge graphs.

However, the "sparse" version of Max Clique (where the number of edges is linear in the number of nodes) looks much easier: for one, Max Clique on $O(n)$-edge graphs is solvable in $2^{O(\sqrt{n})}$ time (see for example Lemma 11.6 in Chapter 11 of Fomin and Kratsch [FK10]). Note that ETH implies that there is no $2^{o(\sqrt{n})}$ time algorithm; in general, known reductions from SAT (and its variants) to Max Clique blow up the instance size by a *quadratic* factor. For this reason, the following problem appears to be open:

**Problem 4:** *Show that Max Clique on $m$-edge graphs has no algorithm running in both $m \cdot poly(\log m)$ time and $O(\log m)$ space.*

Gurevich and Shelah [GS90] proved that the lower bound holds when the input is accessed on a Turing machine tape (with at most poly$(\log n)$ read-only tape heads that move back and forth over the input,

each reading one bit per step). Perhaps their lower bound can be generalized to resolve the problem for random-access models of computation?

Rossman [Ros08] has shown that for small enough $\ell$, the $\ell$-Clique problem requires $\mathsf{AC}^0$ circuits of size at least $n^{\ell/4}$, where $n$ is the number of nodes. This implies a lower bound of $m^{\ell/8}$. If this lower bound could be extended to hold for $\ell > m^\delta$ for some constant $\delta > 0$, then by a folklore translation of time-$t$ log-space computations into depth-$d$ size-$2^{\tilde{O}(t^{1/(d-1)})}$ circuits (see for instance Lemma 7 in Gutfreund and Viola [GV04]), Problem 4 would be answered... but we would have also proved $\mathsf{LOGSPACE} \neq \mathsf{NP}$(!).

**Randomized Algorithms?** Although deterministic linear-time and log-space algorithms for SAT have been ruled out, randomized algorithms are another story. Diehl and Van Melkebeek [DvM06] proved a lower bound for quantified Boolean formulas with two alternations, but the following remains open:

> **Problem 5:** *Show that SAT on $n$-bit instances has no randomized algorithm running in both $n \cdot poly(\log n)$ time and $O(\log n)$ space with one-sided error.*

Here, the "one-sided error" needs to be in the satisfiable case, i.e., there may be an error in the algorithm's output when the input formula is satisfiable, but there's no error when the formula is unsatisfiable. Note that this is the error condition satisfied by every randomized (incomplete) SAT solver. If the one-sided error condition is placed on the unsatisfiable side (so there is no error when formulas are satisfiable), non-trivial lower bounds are known (see Diehl and Van Melkebeek [DvM06]).

As Diehl and Van Melkebeek point out, this problem may seem not so hard at first: why couldn't we assume SAT has such an algorithm, then apply Nisan's deterministic simulation ($\mathsf{RL} \subseteq \mathsf{SC}$ [Nis94]) to obtain a contradiction with known deterministic time-space lower bounds for SAT? The trouble is that Nisan's simulation yields a large polynomial running time, larger than known time lower bounds for deterministically solving SAT.

The main difficulty in attacking Problem 5 seems to be that we do not know enough interesting consequences of randomized super-fast SAT algorithms. Fortnow's approach to SAT lower bounds (and all others after his) derives a contradiction, by using an assumed SAT algorithm to remove alternations from certain alternating machines which simulate log-space computations. We do not know how to effectively remove alternations when we assume a *randomized* algorithm for SAT, in such a way that we can derive a contradiction.

# References


[AK10] Eric Allender and Michal Koucký. Amplifying lower bounds by means of self-reducibility. *J. ACM*, 57(3), 2010.

[And87] Alexander E. Andreev. On a method for obtaining more than quadratic effective lower bounds for the complexity of $\pi$-schemes. *Moscow Univ. Math. Bull.*, 42:63–66, 1987.

[BW15] Samuel R. Buss and Ryan Williams. Limits on alternation trading proofs for time-space lower bounds. *Computational Complexity*, 24(3):533–600, 2015.

[DvM06] Scott Diehl and Dieter van Melkebeek. Time-space lower bounds for the polynomial-time hierarchy on randomized machines. *SIAM J. Comput.*, 36(3):563–594, 2006.

[FK10] Fedor V. Fomin and Dieter Kratsch. *Exact Exponential Algorithms.* Texts in Theoretical Computer Science. An EATCS Series. Springer, 2010.

[For00] Lance Fortnow. Time-space tradeoffs for satisfiability. *J. Comput. Syst. Sci.*, 60(2):337–353, 2000.

[GS90] Yuri Gurevich and Saharon Shelah. Nondeterministic linear-time tasks may require substantially nonlinear deterministic time in the case of sublinear work space. *J. ACM*, 37(3):674–687, 1990.

[GV04] Dan Gutfreund and Emanuele Viola. Fooling parity tests with parity gates. In *RANDOM-APPROX*, pages 381–392, 2004.

[Hås98] Johan Håstad. The shrinkage exponent of De Morgan formulae is 2. *SIAM J. Comput.*, 27:48–64, 1998.

[HMP+93] András Hajnal, Wolfgang Maass, Pavel Pudlák, Mario Szegedy, and György Turán. Threshold circuits of bounded depth. *J. Comput. Syst. Sci.*, 46(2):129–154, 1993.

[IPZ01] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.

[KW16] Daniel M. Kane and Ryan Williams. Super-linear gate and super-quadratic wire lower bounds for depth-two and depth-three threshold circuits. In *STOC*, pages 633–643, 2016.

[Nis94] Noam Nisan. RL <= SC. *Computational Complexity*, 4:1–11, 1994.

[Ros08] Benjamin Rossman. On the constant-depth complexity of k-clique. In *STOC*, pages 721–730, 2008.

[vM07] Dieter van Melkebeek. *A survey of lower bounds for satisfiability and related problems*, volume 7. Now Publishers Inc, 2007.

[Vol99] Heribert Vollmer. *Introduction to Circuit Complexity - A Uniform Approach.* Texts in Theoretical Computer Science. An EATCS Series. Springer, 1999.

[Wil08] R. Ryan Williams. Time-space tradeoffs for counting NP solutions modulo integers. *Computational Complexity*, 17(2):179–219, 2008.

[Wil13] Ryan Williams. Alternation-trading proofs, linear programming, and lower bounds. *TOCT*, 5(2):6, 2013.