Connecting SAT Algorithms and Complexity Lower Bounds

Ryan Williams

IBM Almaden Research Center

Abstract. I will describe prior and current work on connecting the art of finding good satisfiability algorithms with the art of proving complexity lower bounds: proofs of limitations on what problems can be solved by good algorithms. Surprisingly, even minor algorithmic progress on solving the circuit satisfiability problem faster than exhaustive search can be applied to prove strong circuit complexity lower bounds. These connections have made it possible to prove new complexity lower bounds that had long been conjectured, and they suggest concrete directions for further progress.

Recent work has uncovered interesting connections between the satisfiability problem and complexity theory. Let $\mathcal{C}$ be a generic class of Boolean circuits that obey basic properties. Examples of possible $\mathcal{C}$ are the following, listed in increasing order of computational power:

- $\mathsf{AC}^0[m]$ is the class of constant-depth, polynomial-size circuits with unbounded fan-in MODm, AND, and OR gates. (A MODm gate outputs 1 iff the sum of its inputs is divisible by m.)
- ACC is the union over all m of the classes $\mathsf{AC}^0[m]$.
- TC^0 is the class of constant-depth, polynomial-size circuits with unbounded fan-in MAJORITY gates and NOT gates.
- NC^1 is the class of polynomial-size Boolean formulas over the connectives AND, OR, and NOT.
- $\mathsf{P/poly}$ consists of arbitrary polynomial-size Boolean circuits with bounded fan-in AND and OR gates, and NOT gates.

For each such class $\mathcal{C}$, we may define a corresponding $\mathcal{C}$-SAT problem: *given a generic circuit from the class $\mathcal{C}$, is it satisfiable?* Very little is known about the worst-case time complexity of this problem, even when $\mathcal{C}$ is the class of formulas in conjunctive normal form. For example, it is open whether the CNF-SAT problem can be solved in $O(1.9^n)$ time on CNF formulas with n variables and $n^{O(1)}$ clauses.

It turns out that understanding the time complexity of $\mathcal{C}$-SAT is closely related to the problem of simulating computations within the class $\mathcal{C}$. If the worst case $O(2^n)$ time bound for $\mathcal{C}$-SAT can be only slightly improved in some situations, then obstructions *against* $\mathcal{C}$-circuits can be proved. That is, somewhat weak algorithms for solving SAT on interesting circuits can be turned into strong

complexity lower bounds for solving other problems with these interesting circuits. More formally:

> **Theorem.** [1, 2] There is a $c > 0$ such that, if $\mathcal{C}$-SAT can be solved on circuits with $n + c \log n$ inputs and n^k size in $O(2^n/n^c)$ time for every k, then the class NEXP (Nondeterministic Exponential Time) contains languages that cannot be recognized with non-uniform $\mathcal{C}$ circuits of polynomial size.

Intuitively, the theorem states that the difficulty faced by researchers who design fast algorithms for verification of certain kinds of circuits is related to the difficulty of proving that certain problems *can't* be efficiently solved with these kinds of circuits.

Why might such a theorem be true? One intuition is that the existence of a faster $\mathcal{C}$-SAT algorithm shows us a *weakness* in representing computations with circuits from $\mathcal{C}$. The class $\mathcal{C}$ is *not* like a set of black boxes: these circuits cannot hide a satisfying input so easily. Instead, there exists a way to analyze the circuit more efficiently, finding a satisfying input or concluding there is none. Another equally valid intuition is that the existence of a faster SAT algorithm for $\mathcal{C}$ highlights a *strength* of algorithms that run in less-than-2^n time: they can solve nontrivial satisfiability problems.

Summing up, intuition says that a faster SAT algorithm for $\mathcal{C}$ simultaneously shows "less-than-2^n algorithms are strong" and "$\mathcal{C}$-circuits are weak". This gives some hint as to how we might separate the two notions, and prove that some function in nondeterministic exponential time cannot be solved with small (polynomial size) $\mathcal{C}$ circuits. (Warning: the actual proof is much more complicated than this, and proceeds by contradiction. We assume the $\mathcal{C}$-SAT algorithm exists, and that every language in NEXP has small $\mathcal{C}$ circuits. These two assumptions are woven together in nontrivial and unexpected ways, and eventually we derive a contradiction to a previously known complexity lower bound. Hence if the desired SAT algorithm exists, then the small $\mathcal{C}$ circuits cannot exist.)

New circuit complexity lower bounds have recently been proved, building on these connections. In particular, circuit size lower bounds for the class ACC have been proved by designing new algorithms for satisfiability of ACC circuits [2]. It is anticipated that further progress in complexity lower bounds will be made by studying the complexity of satisfiability.

References

1. R. Williams. Improving exhaustive search implies superpolynomial lower bounds. In *ACM Symposium on Theory of Computing*, 231–240, 2010.
2. R. Williams. Non-uniform ACC circuit lower bounds. To appear in *IEEE Conference on Computational Complexity*, 2011.