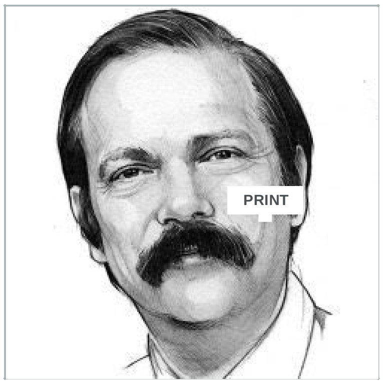
COMMUNICATIONS OF THE ACM

Boolean Satisfiability: Theory and Engineering

By Moshe Y. Vardi

Communications of the ACM, Vol. 57 No. 3, Page 5
10.1145/2578043
Comments



The Boolean Satisfiability Problem (SAT, for short) asks whether a given Boolean formula, with Boolean gates such as AND and NOT, has some assignment of 0s and 1s to its input variables such that the formula yields the value 1. SAT has been a problem of central importance in computer science since Stephen Cook proved its *NP*-completeness in 1971. To resolve the *P* vs. *NP* question, one of the most outstanding open questions in computer science and mathematics, one would have to show whether SAT is or is not in *P*, that is, whether it can be or cannot be solved in polynomial time.

At the same time, SAT is a paradigmatic constraint-satisfaction problem with numerous applications, including hardware and software design, operations research, bioinformatics, and more. Most modern-day SAT-solving algorithms are based on the work of Martin Davis and Hilary Putnam in a 1958 technical report written for the National Security Agency. While we still seem to be quite far from resolving the questions about the computational complexity of SAT, progress on the engineering side has been nothing short of spectacular. Modern SAT solvers routinely solve industrial SAT instances with millions of variables, and are being used by hardware and software designers on a daily basis.

The gap between the impressive progress on the engineering side and very slow progress on the theory side led to an unusual workshop, titled "Theoretical Foundations of Applied SAT Solving," held last January at the Canadian Banff International Research Station for Mathematical Innovation and Discovery (see **http://www.birs.ca/events/2014/5-day-workshops/14w5101 (http://www.birs.ca/events/2014/5-day-workshops/14w5101)**). The workshop brought together leading SAT-complexity theorists and SAT-solving engineers, two groups that rarely meet together and barely speak the same technical language. The hope was that exposing theoreticians and engineers to state-of-the-art developments in SAT theory and engineering would narrow the chasm between these communities.

Of course, one cannot expect robust bridges between two distinct technical communities to be erected in one week, but that should not diminish the tremendous value of such bridge-building workshops. While it may take years to see if this workshop will bear concrete fruits, some fundamental research challenges already emerged.

The most fundamental challenge to emerge goes to the very heart of computational complexity theory. This theory is based typically on worst-case complexity analysis, which focuses on instances that are the most difficult to solve. Worst-case complexity analysis has proven to be quite tractable mathematically, much more, than say average-case complexity analysis. It also seems intuitive from a practical point of view; for example, a worst-case upper bound for an algorithm offers an absolute upper bound on its running time in practice. Thus, worst-case analysis is the standard approach in complexity theory. What has become clear, however, and also became painfully evident at the SAT workshop, is that worst-case analysis actually sheds very little light on the behavior of algorithms on real-life instances. For example, theorists have demonstrated that current SAT-solving algorithms must take exponential time to solve certain families of SAT instances. Practitioners simply shrug at such bounds, while they continue to apply their solvers to very large but practically solvable SAT instances. One role of theory is to provide guidance to engineering, but worst-case (and average-case) complexity seems to offer little guidance for problems that are difficult in theory but feasible in practice. What is needed is a new computational complexity model, which will capture better the concept of "complexity in practice."

A challenge of a different nature applied to the manner in which SAT engineers conduct their research. Current SAT engineering research is intensely heuristic. Researchers devise new heuristics and test their effectiveness on various benchmark suites. An annual SAT competition serves as a powerful motivator, as winning a track of the competition is a path to quick professional recognition. It is impossible to deny this style of research has served the community well, leading to such dramatic progress in SAT solving that researchers refer, jokingly, to "Moore's Law for SAT." At the same time, this heuristic approach lacks science, not only theoretical science but also empirical science. The reality is we have no understanding of why the specific sets of heuristics employed by modern SAT solvers are so effective in practice. Furthermore, we do know there are problem areas where SAT solvers are less effective, but we have no idea why. For the SAT revolution to continue unabated, we must focus also on understanding, not only on benchmarking.

So the bottom line of the workshop is that we need better theory and better engineering. Our work is cut out for us!

Moshe Y. Vardi, EDITOR-IN-CHIEF

No entries found