Schrödinger's SAT: Generalizing Quantum Bogosort to Prove P = NP Under Many-Worlds Quantum Mechanics

Melody "`boringcactus`" Horn

ABSTRACT

Quantum bogosort is a well-known variant of bogosort that exploits the quantum nature of the universe to sort a list in linear time under the many-worlds interpretation of quantum mechanics. We generalize this algorithm to solve the Boolean satisfiability problem in $O(n)$ time. The Boolean satisfiability problem is the original NP-complete problem; as such, this proves that P = NP. This destroyes the RSA cryptosystem.

KEYWORDS

satisfiability, time complexity, revolutions in our understanding of computing, unsolved problems in millennium prize eligibility

1 INTRODUCTION

The Boolean satisfiability problem was the first problem proved to be NP-complete [1, 4]. This result serves as the foundation for all of complexity theory.

Bogosort is a randomized list sorting algorithm that runs in average-case $O(n!)$ time [2]. Quantum bogosort is an adaptation of bogosort that explores all random options simultaneously in different universes and therefore sorts the list in $O(n)$ time [3].

2 PRIOR ART

The bogosort algorithm given in [2] sorts an array a with n elements as follows:

Algorithm 1 Bogosort

```
1: procedure BOGOSORT(a)
2:     while a[1...n] is not sorted do
3:         randomly permute a[1...n]
4:     end while
5: end procedure
```

The quantum bogosort algorithm given in [3] may be formalized analogously as follows:

Algorithm 2 Quantum Bogosort

```
1: procedure QUANTUM-BOGOSORT(a)
2:     randomly permute a[1...n]
3:     if a[1...n] is not sorted then
4:         destroy the entire universe
5:     end if
6: end procedure
```

So long as the random numbers in step 2 are random at a quantum level, the many-worlds interpretation of quantum mechanics indicates that there will be a world where each random permutation is chosen. As such, step 4 ensures that only the worlds where the correct random permutation was chosen continue to exist. Since steps 2 and 3 can run in $O(n)$ time, and step 4 is independent of n and therefore runs in $O(1)$ time, quantum bogosort will sort the array a in $O(n)$ time.

3 METHODS

The Boolean satisfiability problem can be formalized as follows: given some Boolean formula $\Phi(x_1, \ldots, x_n)$ on n variables, find a truth assignment $(x_1, \ldots, x_n) = (T, \ldots, F)$ such that $\Phi(x_1, \ldots, x_n)$ is true, if it exists.[1]

To solve this problem, we present the following algorithm:

Algorithm 3 Schrödinger's SAT

```
1: procedure SCHRÖDINGER'S-SAT(Φ)
2:     for i ← 1, n do
3:         Randomly guess either x_i ← T or x_i ← F
4:     end for
5:     if ¬Φ(x_1, ..., x_n) then
6:         destroy the entire universe
7:     end if
8:     return (x_1, ..., x_n)
9: end procedure
```

As with quantum bogosort, if the guess in step 3 is random at a quantum level, there will be a world for each value, and so by step 4 there is a world for every possible truth assignment. As such, by step 8 we have found a satisfying truth assignment. (Sufficiently bored or curious readers may wish to implement this algorithm and run it on $\Phi(x_1) = x_1 \wedge \neg x_1$.)

Since there are n guesses made, each guess takes $O(1)$ time, and the formula can be evaluated in $O(n)$ time, this algorithm finds a satisfying assignment in $O(n)$ time, demonstrating that Boolean satisfiability is in P and therefore that P = NP.

ACKNOWLEDGMENTS

To Joe.

REFERENCES

[1] Stephen A. Cook. 1971. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing - STOC '71*. ACM Press, Shaker Heights, Ohio, United States, 151–158. https://sci-hub.se/10.1145/800157.805047

[2] Hermann Gruber, Markus Holzer, and Oliver Ruepp. 2007. Sorting the Slow Way: An Analysis of Perversely Awful Randomized Sorting Algorithms. In *Fun with Algorithms*, Pierluigi Crescenzi, Giuseppe Prencipe, and Geppino Pucci (Eds.). Vol. 4475. Springer Berlin Heidelberg, Berlin, Heidelberg, 183–197. https://sci-hub.se/10.1007/978-3-540-72914-3_17 Series Title: Lecture Notes in Computer Science.

[3] The Other Tree. 2009. Quantum Bogosort. *mathNEWS* 111, 3 (Oct. 2009), 13. https://mathnews.uwaterloo.ca/wp-content/uploads/2014/08/v111i3-compressed.pdf

[1] As is standard practice, we handwave away the difference between the decision problem and the search problem.

[4] B.A. Trakhtenbrot. 1984. A Survey of Russian Approaches to Perebor (Brute-Force Searches) Algorithms. *IEEE Annals of the History of Computing* 6, 4 (Oct. 1984), 384–400. https://sci-hub.se/10.1109/MAHC.1984.10036