A Polynomial Time Algorithm for 3SAT

ROBERT QUIGLEY

This paper includes the presentation of a polynomial time algorithm to solve 3SAT. It begins with the problem definition and format of various parts of the problem. It then offers various lemmas of important aspects of the 3SAT problem along with their corresponding proofs. Next, it describes the algorithm that uses these lemmas to solve 3SAT. And finally, it proves that the algorithm works for any general case of 3SAT. The algorithm relies on the fact that an instance of 3SAT is unsatisfiable iff contradicting 1-terminal clauses can be derived in polynomial time.

Additional Key Words and Phrases: test1, test2

ACM Reference Format:
Robert Quigley. 2024. A Polynomial Time Algorithm for 3SAT. 1, 1 (January 2024), 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

TODO: find who the heck wrote '3SAT is NP-complete' read https://dl.acm.org/doi/10.1145/800157.805047

1 DEFINITIONS

Definition 1.1. Terminals: symbols used in the 3SAT problem that can be assigned a value of either 0 or 1, True or False, or any other binary assignment. They usually take the form x_i where i is a natural number.

Definition 1.2. Clauses: a set of terminals combined by logical or operators. Each terminal can also be negated. Clauses usually take the form $(x_i \vee \neg x_j \vee x_k)$ where $x_i \neq x_j \neq x_k$

Definition 1.3. (3SAT) Instance: a set of any number of clauses combined by logical and operators. Note that entire clauses may not be negated. Terminals may not repeat within a clause, but they are free to repeat between clauses. Instances usually take the form: $(x_i \vee \neg x_j \vee x_k) \wedge (x_l \vee x_m \vee x_n)$

Definition 1.4. Assignment: A list of values in which each value represents either True or False such that each item in the list corresponds to a terminal and all terminals are assigned a value.

Definition 1.5. Satisfying Assignment: An assignment, A, is said to satisfy the instance, if applying A will make the instance evaluate to True.

Definition 1.6. The 3SAT Problem: Given a 3SAT instance, does there exist a satisfying assignment?

Definition 1.7. Blocking an Assignment: An assignment, A, is said to be blocked if, given a clause, C, there is no way that A can satisfy the instance.

Author's address: Robert Quigley, robertquigley21@gmail.com.

Definition 1.8. TODO: define a more 'specifically in polytime' type of implication Implication: A clause, C, is said to imply another clause, D, if all assignments blocked by D are also blocked by C

Definition 1.9. Given Clauses: clauses which were given in the original instance.

Definition 1.10. Implied Clauses: clauses which are implied by the clauses in the original instance.

Definition 1.11. k-terminal (k-t) clause: a clause is described as a k-terminal or a k-t clause if there are k terminals in the clause

2 REFORMATTING

Since there are a lot of constant characteristics about an instance of 3SAT, we can remove most of them to allow ourselves to focus only on what changes from instance to instance. A list of unchanging characteristics follows:

- the symbol x
- logical and operators
- logical or operators

The only difference between instances, therefore, is the subscript of the terminal. The following items will be changed to improve compatibility with python code: *possible footnote to describe how to represent it as list of lists

- parentheses will become square brackets
- negation symbols will become minus signs
- an instance may be surrounded with square brackets to show it is a list of lists

For example, the instance:
$(\neg x_a \vee x_b \vee x_c) \wedge (x_a \vee x_d \vee x_e)$
will be written as:
$[[-a, b, c], [a, d, e]]$

3 LEMMAS

3.1 Lemma A

3.2 Lemma B

3.3 Lemma C

3.4 Lemma D

3.5 Lemma E

4 ALGORITHM

5 TIME COMPLEXITY ANALYSIS

6 PROOF OF CORRECTNESS

7 CONCLUSION

TODO:

- reorder definitions