3-SAT Solver

Najib Ishaq
najib_ishaq@zoho.com

ABSTRACT

Abstract is written last.

CCS CONCEPTS

• **Mathematics of computing** → **Solvers**.

KEYWORDS

P vs NP

ACM Reference Format:
Najib Ishaq. 2020. 3-SAT Solver. In *Proceedings of ACM Conference (Conference'20).* ACM, San Diego, CA, USA, 2 pages. https://doi.org/xxxx

1 INTRODUCTION

Introductions are in order.

2 DEFINITIONS

We set up the problem with some definitions.

3-SAT Instance

A 3-SAT *instance* of size (m, k) is a conjunction of m *clauses*, where each *clause* is a disjunction of three *literals*, where *literal* is either a *variable* or the *negation of a variable*, where each *variable* is taken from a set of k *variables*.

$$\Phi \coloneqq C_1 \wedge C_2 \ldots \wedge C_m$$
$$where\ C_i \coloneqq (x \vee y \vee z)\ \forall i \in \{1 \ldots m\}$$
$$where\ x, y, z \in \{x_1 \ldots x_k, \neg x_1 \ldots \neg x_k\}$$

Satisfiability

If an *instance* has at least one set of mappings from its *variables* to truth values then it is *satisfiable*, otherwise it is *unsatisfiable*.

$$\Phi\ is\ satisfiable \ \Leftrightarrow\ \exists\ \mathcal{X} \mid \Phi(\mathcal{X}) = T$$
$$\Phi\ is\ unsatisfiable \ \Leftrightarrow\ \neg\exists\ \mathcal{X} \mid \Phi(\mathcal{X}) = T$$

If after assigning truth values to some, but not all, *variables* and evaluating the *instance* we find that there are some *clauses* left whose truth value is ambiguous, we say that the *instance* is, as yet, *undecided*.

Certificate

A *certificate* is a set of mappings from each *variable* x_i to a truth value such that the set of assignments *satisfies* a given *instance*.

$$\mathcal{X} :\ x_i \mapsto \{T,\ F\}\ \forall i \in \{1 \ldots k\} \mid \Phi(\mathcal{X}) = T$$

Min-Certificate

A *certificate* may not need to specify a mapping for **every** *variable* in an *instance*. Some *instances* can be *satisfied* by assigning a mapping for some, but not all, of the *variables* in that *instance*.

A *min-certificate* is a special case of a *certificate* with the property that removing even one mapping from the *certificate* leads to the *instance* being left *undecided*.

For example:

$$\phi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_2 \wedge \neg x_3 \wedge x_4)$$

can be *satisfied* by $\mathcal{X} = \{x_2 \mapsto F\}$. This leaves 3 variables unmapped, which leads to 2^3 full *certificates*. $\mathcal{X}$ is a *min-certificate* for the 3-SAT instance Φ.

There are other possible *min-certificates* for this example. For example:

- $\{x_1 \mapsto T,\ x_3 \mapsto F\}$ which leads to 2^2 full *certificates*.
- $\{x_3 \mapsto T,\ x_4 \mapsto T\}$ which leads to 2^2 full *certificates*.

For the remainder of this paper, the term *certificate* will be overloaded with this definition of a *min-certificate*.

Child Instances

Given an *instance* Φ with more than one *clause*, we define two *instances*, ϕ_{left} and ϕ_{right}, whose clauses are disjoint subsets of the set of *clauses* from Φ, such that $\Phi = \phi_{left} \wedge \phi_{right}$.

ϕ_{left} and ϕ_{right} are *child instances* with Φ as the *parent instance*. ϕ_{left} and ϕ_{right} are *siblings* to each other.

We have:

$$\phi_{left},\ \phi_{right} \subset \Phi \mid \Phi = \phi_{left} \wedge \phi_{right},\ \phi_{left} \cap \phi_{right} = \emptyset$$

Child Certificates

If Φ is *satisfiable*, then so are ϕ_{left} and ϕ_{right}. In other words, if Φ has a *certificate*, then each *child instance* has at least one *certificate*. The *certificates* for *child instances* are called *child certificates*.

Conference'20, August 2020, San Diego, CA, USA

https://doi.org/xxxx

Given Φ and χ, we can find a *child certificate*, χ_{child}, for ϕ_{child} by taking the subset from χ of mappings of the *variables* that are present in ϕ_{child}.

Certificate Compatibility

Given *sibling instances*, ϕ_{left} and ϕ_{right}, and their respective *certificates*, χ_{left} and χ_{right}, we define *certificate compatibility* in the following way.

$$\chi_{left} \textit{ is compatible with } \phi_{right}$$
$$\Leftrightarrow \phi_{right}(\chi_{left}) \textit{ is either True or Undecided}$$

3 SUB-PROBLEMS

We define some sub-problems and consider how to solve each of them.

Finding Parent Certificates

Given *sibling instances* and their *certificates*, $(\phi_{left},\ \chi_{left})$ and $(\phi_{right},\ \chi_{right})$, such that χ_{right} is compatible with ϕ_{left} and χ_{left} is compatible with ϕ_{right}, we can find the *parent certificate*, χ by taking the union of χ_{left} and χ_{right}.

4 THE SOLVER

We put the pieces together into the full algorithm.

5 ANALYSIS

We analyze the time and space complexity of the proposed algorithm.

6 CONCLUSIONS

We learned some things.