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, 3 pages. https://doi.org/X

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, n) 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 n *variables*.

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

A 2-SAT instance like a 3-SAt instance except that each clause has two literals instead of three.

Unit Clauses and Pure Literals

A *unit clause* is a clause with only one literal. Such a clause can always be satisfied by setting that literal to T.

A *pure literal* is one whose negation never appears in the instance. All clauses containing a *pure literal* can be satisfied by setting that literal to T.

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

https://doi.org/X

Certificate

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

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

Satisfiability

If an instance has at least one certificate then it is *satisfiable*, otherwise it is *unsatisfiable*.

$$\Phi\ is\ satisfiable\ \Leftrightarrow\ \exists\ \chi \mid \Phi(\chi) = T$$
$$\Phi\ is\ unsatisfiable\ \Leftrightarrow\ \neg\exists\ \chi \mid \Phi(\chi) = T$$

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*.

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 $\chi = \{x_2 \mapsto F\}$. This leaves 3 variables unmapped, which leads to 2^3 full certificates. χ 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} such that every clause from Φ in either in ϕ_{left} or ϕ_{right} but not in both.

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

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

Child Certificates

If Φ is satisfiable then so are its child instances 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*.

Given Φ and $\mathcal{X}$, we can find a child certificate, $\mathcal{X}_{child}$, for ϕ_{child} by taking the subset of $\mathcal{X}$ of mappings of the variables that are present in ϕ_{child}.

Compatibility

Given two instances, ϕ_1 and ϕ_2, and their respective certificates, $\mathcal{X}_1$ and $\mathcal{X}_2$, we define *certificate compatibility* in the following way.

$$\begin{aligned} &\mathcal{X}_1 \textit{ is compatible with } \phi_2 \\ \Leftrightarrow\ &\phi_1(\mathcal{X}_2) \textit{ is either True or Undecided} \end{aligned}$$

We can also define *compatibility* among two certificates. If every variable that is assigned a truth value in both $\mathcal{X}_1$ and $\mathcal{X}_2$ is assigned the **same** truth value in both $\mathcal{X}_1$ and $\mathcal{X}_2$, then $\mathcal{X}_1$ and $\mathcal{X}_2$ are *compatible* with each other.

$$\begin{aligned} &\mathcal{X}_1 \textit{ is compatible with } \mathcal{X}_2 \\ \Leftrightarrow\ &\forall x_j \in (\mathcal{X}_1 \cap \mathcal{X}_2) \\ &\textit{we have } \mathcal{X}_1(x_j) \equiv \mathcal{X}_2(x_j) \end{aligned}$$

Note that if $\mathcal{X}_1$ is compatible with ϕ_2 and $\mathcal{X}_2$ is compatible with ϕ_1 then $\mathcal{X}_1$ is compatible with $\mathcal{X}_2$. From two such instances and certificates, we note that $\mathcal{X}_1 \cup \mathcal{X}_2$ is a certificate for $\phi_1 \wedge \phi_2$.

Base-Case Instance

A *base-case instance*, ϕ_{base}, is a special case of an instance in which all clauses have a variable, or its negation, in common.

$$\begin{aligned} &let\ \phi_{base} := C_1 \wedge C_2 \wedge \ldots \wedge C_m \\ &s.t.\ either\ x_j \in C_i\ or\ \neg x_j \in C_i \\ &for\ each\ i \in \{1 \ldots m\} \\ &for\ some\ j \in \{1 \ldots n\} \end{aligned}$$

3 SUB-PROBLEMS

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

Partitioning an Instance

Given an instance Φ with variables $\{x_1 \ldots x_n\}$, we can partition Φ into two child instances about the variable x_i by whether or not the clauses in Φ contain the literals x_i or $\neg x_i$. Without loss of generality, we can have ϕ_{left} be the conjunction of clauses that each **contain** the literals x_1 or $\neg x_1$, and ϕ_{right} be the conjunction of clauses that each **do not contain** the literals x_1 or $\neg x_1$.

Note that ϕ_{left} is a base-case instance and ϕ_{right} is an instance of the same form as Φ but with a smaller size $(m',\ n')$ with $m' \leq m - 1$ and $n' \leq n - 1$. If we always remove unit clauses and pure literals, then $m' \leq m - 1$ and $n' \leq n - 2$.

Solving a Base-Case Instance

Given a base-case instance ϕ with the common *variable* x_j, we can further partition ϕ into two child instances ϕ^1 and ϕ^2 such that ϕ^1 contains only those clauses that contain the literal x_j and ϕ^2 contains only those clauses that contain the literal $\neg x_j$.

ϕ is satisfiable if and only if ϕ^1 and ϕ^2 are satisfiable and ϕ^1 has at least one certificate compatible with ϕ^2 and ϕ^2 has at least one certificate compatible with ϕ^1.

We will find all certificates for ϕ^1 and ϕ^2. For ϕ^1, one certificate is $\{x_j \mapsto True\}$. In order to find the other certificates, we start with $\{x_j \mapsto False\}$ and reduce ϕ^1 into a 2-SAT instance. We then find all certificates for that 2-SAT instance. By symmetry, ϕ^2 can be satisfied by $\{x_j \mapsto False\}$ or by starting with $\{x_j \mapsto True\}$ and finding all certificates for the resulting 2-SAT instance.

Finding Parent Certificates

Given sibling instances and all their respective certificates, $(\phi_{left},\ \{\mathcal{X}_{left}\})$ and $(\phi_{right},\ \{\mathcal{X}_{right}\})$, we discard any certificate for ϕ_{left} that is not compatible with ϕ_{right} and we discard any certificate for ϕ_{right} that is not compatible with ϕ_{left}. We then take the two sets of mutually compatible certificates and obtain the set of parent certificates by performing pairwise unions between certificates from either set.

Note that Φ is satisfiable if and only if $\exists\ \mathcal{X}_{left}, \mathcal{X}_{right}$ such that $\mathcal{X}_{left}$ and $\mathcal{X}_{right}$ are mutually compatible.

4 THE SOLVER

We piece together the sub-problems into the complete algorithm. We will assume that we remove unit clauses and pure literals at each appropriate step.

2-SAT Solver

Consider a 2-SAT instance, Φ of size $(m,\ n)$. Partition Φ about some choice of variable, x_j, to obtain ϕ_{left} and ϕ_{right}. ϕ_{left} is a base-case instance, while ϕ_{right} is a smaller 2-SAT instance which we will solve recursively.

Note that if Φ is satisfiable, then ϕ_{left} and ϕ_{right} are also satisfiable and each has at least one certificate compatible with the other.

Now consider a base-case 2-SAT instance, ϕ. Partition ϕ into ϕ_1 and ϕ_2 such that ϕ_1 has only those clauses that contain x_j and ϕ_2 has only those clauses that contain $\neg x_j$.

There are at most two possible certificates for each of ϕ_1 and ϕ_2. For ϕ_1, let us call these certificates $\mathcal{X}_{1\alpha}$ and $\mathcal{X}_{1\beta}$.

$\chi_{1\alpha}$ and $\chi_{1\beta}$ are defined as follows:

$$\chi_{1\alpha} \coloneqq \{x_j \mapsto T\}$$
$$\chi_{1\beta} \coloneqq \{x_j \mapsto F,\ \textit{all other literals} \mapsto T\}$$

$\chi_{1\beta}$ exists if and only if all variables, other than x_j, are present as pure literals, i.e. if a variable is present then its negation is absent.

By symmetry, we have:

$$\chi_{2\alpha} \coloneqq \{x_j \mapsto F\}$$
$$\chi_{2\beta} \coloneqq \{x_j \mapsto T,\ \textit{all other literals} \mapsto T\}$$

for ϕ_2. Note that $\chi_{1\alpha} \subset \chi_{2\beta}$ and $\chi_{2\alpha} \subset \chi_{1\beta}$.

If ϕ is satisfiable, then ϕ_1 and ϕ_2 are also satisfiable and each has at least one certificate compatible with the other. $\chi_{1\alpha}$ is not compatible with $\chi_{2\alpha}$. Therefore either one or both of the following must be true:

$$\exists!\ \chi_{1\beta}\ \textit{compatible with}\ \phi_2$$
$$\exists!\ \chi_{2\beta}\ \textit{compatible with}\ \phi_1$$

Therefore either one or both of $\chi_{1\beta}$ and $\chi_{2\beta}$ are certificates for ϕ.

Following this procedure gives us all certificates for each pair of sibling instances. We then discard those certificates that are not compatible with their sibling instance. We then obtain the certificates for the parent instance by performing pairwise unions between the remaining certificates from each sibling.

3-SAT Solver

...

5 ANALYSIS

We analyze the time and space complexity of the proposed algorithm. We will proceed under the assumption that variables and their negations are uniformly distributed among clauses and that no one clause has both a variable and its negation. This eases the analysis but it should hold even if the uniformity assumption is incorrect.

Partitioning an Instance

We expect for ϕ_{left} to have $\frac{1}{n} \cdot m$ clauses from Φ and for ϕ_{right} to be an instance of size $(\frac{n-1}{n} \cdot m,\ n-1)$. In the worst-case, ϕ_{left} will have 2 clauses and ϕ_{right} will be an instance of size $(m-1,\ n-2)$.

Since this partitioning step checks for membership of two literals in each clause, it takes $O(m)$ time.

Solving a Base-Case Instance

...

Finding Parent Certificates

...

Since there are more clauses for each certificate to satisfy, we expect a majority of certificates to be discarded in this step.

2-SAT Solver

...

3-SAT Solver

...

6 CONCLUSIONS

We learned some things.