SAT and Its Variants

Sunghyeon Jo

Seoul National University

Project-Hardness
Sep 27, 2022

We will look...

- many variants of SAT
  - classify which is P or NP-complete (or else?)
- use these variants of SAT to prove many problems NP-complete

Definitions

- Let φ be a Boolean formula.
  1. The terms $x_i, \neg x_i$ are called **literals**.
     x_i is **postive** and $\neg x_i$ is **negative** literal.
  2. φ is in **Conjunctive Normal Form(CNF)** if
     $\varphi = C_1 \wedge ... \wedge C_k$ where C_j's are ORs of literals.
  3. if φ is a CNF formula, C_j's are called **clauses**.
  4. φ is in **Disjunctive Normal Form(DNF)** if
     $\varphi = D_1 \vee ... \vee D_k$ where D_j's are ANDs of literals.
  5. A **Boolean circuit** is a circuit with inputs $x_1, ..., x_n$ and one output.
     it can has three type of gates, AND, OR, and NOT.

Famous results

problem XSAT : given a formula of the form X. is it satisfable?
(ex. CNF SAT)

Theorem

CNF SAT is NP-complete. (Cook71)

Corollary

Circuit SAT is NP-complete.

Famous results

Theorem

DNF SAT is in P.

Given DNF formula φ, consider each conjunction of φ.
If there is a conjuction does not include both x and $\neg x$ for every variable x, then the conjunction can have the value **TRUE** hence φ is satisfiable.
Otherwise, φ is not satisfiable.

Relation between CNF and DNF

Question

Given a CNF formula φ. Is there a DNF formula ψ which is equivalent to φ?

Since $\wedge$ and $\vee$ satisfies distributive property, we can unfold the CNF formula and make it in DNF form like below example:

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

Then why CNF SAT is NP-complete while DNF SAT is P?

Relation between CNF and DNF

Theorem

Any DNF formula that is equivalent to $(x_1 \vee y_1) \wedge (x_2 \vee y_2) \wedge ... \wedge (x_n \vee y_n)$ requires at least 2^n clauses.

Suppose DNF formula φ is equivalent to above CNF.
Then, each clause of φ should have one of x_i and y_i for each i since each clause should be false when $x_i = y_i =$ FALSE and $x_j = y_j =$ TRUE for all $j \neq i$.

On the other hand, the original CNF is equivalent to
$\phi = (x_1 \wedge x_2 \wedge .... \wedge x_n) \vee (x_1 \wedge x_2 \wedge .... \wedge y_n) \vee ... \vee (y_1 \wedge y_2 \wedge .... \wedge y_n)$
(total 2^n clauses). For any two clauses A, B of ϕ, they do not have a relation $A \Rightarrow B$. And for any clause A of φ, we can find a clause B_A of ϕ such that $A \Rightarrow B_A$.

Therefore, any DNF formula equivalent to original CNF has at least same clauses as ϕ.

Variants of SAT that are mostly NP-Complete

From now, we will use SAT to mean CNF-SAT.
There are many variants of SAT, which are based on SAT but adding some restriction on formula or solutions.

Definition

aSAT : every clause(of given formula) has $\leq a$ literals.

Theorem

2SAT is in P.
3SAT is in NP-complete.

Variants of SAT that are mostly NP-Complete

- EaSAT: every clause has **exactly** a literals.
- EUaSAT: EaSAT, every variable within a clause occurs **uniquely**.
- aSAT-b: aSAT, every variable occurs **at most** b times.
- aSAT-Eb: aSAT, every variable occurs **exactly** b times.
- EaSAT-b, EUaSAT-b, EaSAT-Eb, EUaSAT-Eb, etc..

Variants of SAT that are mostly NP-Complete

Theorem

3SAT-3 is in NP-complete.

Claim: 3SAT $\leq_p$ 3SAT-3.

For each variable x occurs $k > 3$ times, change last $k - 1$ occurrence of x to $y_1, y_2, .., y_{k-1}$ and add clauses $x \vee \neg y_1$, $y_i \vee \neg y_{i+1}$, $y_{k-1} \vee \neg x$ at last of the formula.

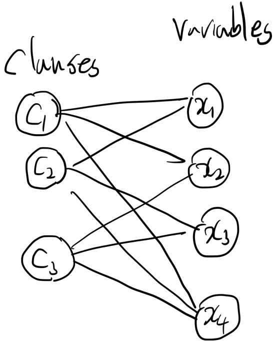
Then new formula is equivalent to original formula and occurrences of x and y_i are at most 3. By repeating this operation, we can make formula in right form of 3SAT-3.

Variants of SAT that are mostly NP-Complete

Theorem

EU3SAT-3 $\in$ P, And all formulas of this form is satisfiable.

Consider a bipartite graph with clauses on the left and variables on the right. clause C and variable x is connected by edge if x or $\neg x$ is contained in C.



$deg(C_i) = 3, deg(x_i) \leq 3.$

By Hall's theorem, there is a matching covers all clauses. Therefore, if C_i is matched with x_{p_i}, we can manipulate x_{p_i}'s value to make all C_i TRUE.

Variants of SAT that are mostly NP-Complete

Definition

GAP aSAT-b: Given formula in the form of aSAT-b which is guaranteed to be either satisfiable or at most $1 - \epsilon$ fraction of its clauses can be simultaneously satisfied.

Determine which is the case.

Theorem

GAP 3SAT-5 is NP-hard.

Variants of SAT that are mostly NP-Complete

- NAE-: For each clause, there should be a literal with value TRUE and a literal with value FALSE.
- 1-IN-: For each clause, exactly one literal should be TRUE.
- Monotone: For each clause, all of its literals are positive or all of its literals are negative.
- Positive: For each clause, all of its literals are positive.

Note that NAE- and 1-IN- restriction is restriction for the solution, **not** for the given CNF.

Example

Monotone 1-IN-3SAT

Instance: A CNF φ is form of 3SAT and Monotone.
Task: Find a solution s.t. every clause of φ have exactly one literal of value TRUE.

Variants of SAT that are mostly NP-Complete

- Monotone 3SAT is NP-Complete.
- Monotone 3SAT-E4 is NP-Complete. (solved in 2018)

Variants of SAT that are mostly NP-Complete

MAX 2SAT: Given 2CNF φ, find maximum number of clauses that can be simultaneously satisfied.

Theorem

MAX 2SAT is NP-Complete.

Corollary

Following problem is NP-Complete: Given 3CNF formula φ. Is there a satisfying assignment s.t. majority of clauses have all three literals TRUE?

Proof : Exercise.

Variants of SAT that are mostly NP-Complete

Theorem

Those 4 Variants of SAT are NP-Complete.

- *1-IN-3SAT*
- *Positive 1-IN-3SAT*
- *NAE-3SAT*
- *Positive NAE-3SAT*

For a clause $(x \vee y \vee z)$,
$(\neg x \vee a \vee b) \wedge (b \vee y \vee c) \wedge (c \vee d \vee \neg z)$ is 1-IN-3SAT satisfiable iff at least one of x, y, z is TRUE.
Therefore, 3SAT $\leq_p$ 1-IN-3SAT.

Schaefer's Dichotomy Theorem

Definition

A **relation** on m variables are a formula on those variables. i.e. a function from $\{0,1\}^m$ to $\{0,1\}$.

Assume that we have an instance of 1-IN-3SAT,
$\varphi = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_3 \vee \neg x_4)$.
Since there is a clause $(x_1 \vee x_2 \vee x_3)$, one out of three should be TRUE and other two should be FALSE.
it is equivalent to a relation
$R_1(x_1, x_2, x_3) = (x_1 \wedge \neg x_2 \wedge \neg x_3) \vee (\neg x_1 \wedge x_2 \wedge \neg x_3) \vee (\neg x_1 \wedge \neg x_2 \wedge x_3)$.
Similarly, Define a relation
$R_2(x_1, x_2, x_3) = (x_1 \wedge \neg x_2 \wedge x_3) \vee (\neg x_1 \wedge x_2 \wedge x_3) \vee (\neg x_1 \wedge \neg x_2 \wedge \neg x_3)$.

Then, the instance φ can be rewritten as $R_1(x_1, x_2, x_3) \wedge R_2(x_1, x_3, x_4)$.

Schaefer's Dichotomy Theorem

Definition

A SAT-type problem is a set of relations $R_1, R_2, ..., R_k$.

Let's consider previous example.

For two relations R_1, R_2,
1-IN-3SAT for $\varphi = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_3 \vee \neg x_4)$ is an instance of SAT-type problem $\{R_1, R_2\}$. However, it is not an instance of $\{R_1\}$.

We will assume none of the R_i are constant in a SAT-type problem.

Schaefer's Dichotomy Theorem

Theorem (Schaefer's Dichotomy Theorem)

Let $R_1, R_2, ..., R_k$ be relations. If any of the following occur, then the SAT-type problem $\{R_1, R_2, ..., R_k\}$ is P. Otherwise, it is NP-Complete.

- $\forall R_i, R_i$*(TRUE, TRUE, .., TRUE)* $=$ *TRUE. (***1-valid** *relation)*
- $\forall R_i, R_i$*(FALSE, FALSE, ..., FALSE)* $=$ *TRUE. (***0-valid** *relation)*
- $\forall R_i, R_i$ *is equivalent to a conjunction of clauses with* ≤ 2 *variables. (***bijunctive** *relation)*
- $\forall R_i, R_i$ *is equivalent to a conjunction of clauses with* ≤ 1 *positive literal (***Horn** *clauses) :* **weakly negative** *relation.*
- $\forall R_i, R_i$ *is equivalent to a conjunction of clauses with* ≤ 1 *negative literal (***Dual-Horn** *clauses) :* **weakly positive** *relation.*
- $\forall R_i, R_i$ *is equivalent to an* **affine** *formula. That is, conjunction of xor-clauses e.g.* $(x_1 \oplus \neg x_2 \oplus x_3) \wedge (x_2 \oplus x_4)$.

Note: the book is wrong

Schaefer's Dichotomy Theorem

It's easy to see each 6 cases are trivially included in P. (Exercise)

...Why all the other cases are in NP-Complete?

.....And why there are only P and NP-Complete problems in SAT-type problems?

Schaefer's Dichotomy Theorem

Lemma

Let
$R(x_1, x_2, x_3) = (x_1 \wedge \neg x_2 \wedge \neg x_3) \vee (\neg x_1 \wedge x_2 \wedge \neg x_3) \vee (\neg x_1 \wedge \neg x_2 \wedge x_3)$
Then 3SAT $\leq_p$ (SAT-type problem $\{R\}$).

R is TRUE when exactly one out of three is TRUE.
Let
$A = R(x, u_1, u_4) \wedge R(y, u_2, u_4) \wedge R(u_1, u_2, u_5) \wedge R(u_3, u_4, u_6) \wedge R(z, u_3, 0)$,
$B = R(x, y, 0)$,
$C = R(x, x, u)$

$\exists_{(u_i)}\ A$ is equivalent to $(x \vee y \vee z)$ and B is equivalent to $x \neq y$,
$\exists_u\ C$ is equivalent to $x = 0$. Therefore, we can make all kinds of clauses with 3 literals by adding additional variables.

Schaefer's Dichotomy Theorem

Let $Rep(R_1, .., R_k)$ be set of relations that can be represented by only $R_1, .., R_k$.
For example, for R in the last slide, $[x \vee y \vee z], [x \neq y], [\neg x] \in Rep(R)$.

Lemma

Let $S = \{R_1, R_2, ..., R_k\}$ be set of relations that is not all of them are 0-valid, and not all of them are 1-valid.
Then, It is guaranteed that $[x \neq y] \in Rep(S)$ or $[x], [\neg x] \in Rep(S)$.
Moreover, If $[x], [\neg x] \notin Rep(S)$, then $Rep(S)$ is complementive (For all $R \in Rep(S)$, $R(x_1, .., x_m) = R(\neg x_1, ..., \neg x_m)$)

if every relation in S is 0-valid or 1-valid, then there is $R_0, R_1 \in S$ s.t. R_0 is 0-valid and not 1-valid, R_1 is 1-valid and not 0-valid. then $[x], [\neg x] \in Rep(S)$.

Schaefer's Dichotomy Theorem

Let R be a relation in S that is neither 0-valid nor 1-valid. Choose some $(x_1, ..., x_m)$ satisfies R. Then replace all variables with value 0 to y_0, value 1 to y_1. Then set of (y_0, y_1) satisfies R can be $\{(0,1)\}$ or $\{(0,1),(1,0)\}$ since it is not 0,1-valid.
Former case, $[x], [\neg x] \in Rep(S)$. Latter case, $[x \neq y] \in Rep(S)$.

Exercise: Prove the additional claim in lemma : If $[x], [\neg x] \notin Rep(S)$, then $Rep(S)$ is complementive.
Note that $[x], [\neg x] \in Rep(S)$ is equivalent to allowing using 0, 1 in relation.

Schaefer's Dichotomy Theorem

Lemma

If $S = \{R_1, R_2, ..., R_k\}$ *contains both* **not** *weakly negative relation and* **not** *weakly positive relation and* $[x], [\neg x] \in Rep(S)$*, then* $[x \neq y] \in Rep(S)$.

Lemma

If R *is not bijunctive, then* $Rep(\{R, [x \neq y], [x \vee y]\})$ *contains the relation "exactly one of x,y,z"*

Lemma

If R *is not affine,*
$[x \vee y], [\neg x \vee y], [x \vee \neg y], [\neg x \vee \neg y] \in Rep(\{R, [x \neq y], [x]\})$.

The proof of above lemmas are exercise: similar to the proof for NP-Completeness of "exactly one of x,y,z" relation.

Schaefer's Dichotomy Theorem

Now we are ready to prove Schaefer's Dichotomy Theorem.

Suppose that $S = \{R_1, R_2, ..., R_k\}$ is not included to six P cases.

Then it has non-affine relation, non-weakly negative relation, non-weakly positive relation, non-bijunctive relation.

Therefore, if $[x], [\neg x] \in Rep(S)$, then $[x \neq y] \in Rep(S)$, hence $[x \vee y] \in Rep(S)$, thus "exactly one of x,y,z" is in $Rep(S)$.

As a result, 3SAT $\leq_p S$.

Schaefer's Dichotomy Theorem

Otherwise, $Rep(S)$ is complementive. If we can use 0, 1 in relations, then it is the same to $[x], [\neg x] \in Rep(S)$ and it results in that we can make every 3-literal clauses by building "exactly one of x,y,z" relation. For any 3-literal clause C, it is possible to make equivalent relation consists of relations in S and using additional variable 0, 1.
Instead of using 0 and 1, use y_0, y_1 and add clause $\wedge(y_0 \neq y_1)$. Since $Rep(S)$ is complementive, it is equivalent to C.
In consequence, 3SAT $\leq_p S$.

A Dichotomy Theorem for Graph Formulas

Definition

A **graph formula** is a Boolean formula where all literals are of the form $E(x, y)$ or $x = y$.
A graph formula $\varphi(x_1, ..., x_n)$ is **satisfiable** if there exists a graph G of vertex set $\{v_1, v_2, ..., v_n\}$ such that $\varphi(v_1, ..., v_n)$ is true in G.

$$\left(\bigwedge_{1\leq i<j\leq 6} \neg(x_i = x_j)\right) \bigwedge \left(\bigwedge_{1\leq i\neq j\leq 6} \neg E(x_i, x_j) \wedge E(x_j, x_i)\right) \bigwedge$$

$$\left(\bigwedge_{1\leq i<j<k\leq 6} \neg(E(x_i, x_j) \wedge E(x_j, x_k) \wedge E(x_k, x_i))\right)$$

:An undirected (2nd term) graph with 6 vertices with no three edges form triangle(3rd term).

A Dichotomy Theorem for Graph Formulas

Let $\Psi = \{\psi_1, \ldots, \psi_n\}$ be a finite set of propositional (Boolean) formulas.

Boolean-SAT(Ψ)
INSTANCE: Given a finite set of variables W and a propositional formula of the form $\Phi = \phi_1 \wedge \cdots \wedge \phi_l$ where each ϕ_i for $1 \le i \le l$ is obtained from one of the formulas ψ in Ψ by substituting the variables of ψ by variables from W.
QUESTION: Is there a satisfying Boolean assignment to the variables of W (equivalently, those of Φ)?

Above is rewritten version of defining SAT-type problem Ψ.
Let's think about graph formula version.

A Dichotomy Theorem for Graph Formulas

Graph-SAT(Ψ)
INSTANCE: Given a set of variables W and a graph formula of the form $\Phi = \phi_1 \wedge \cdots \wedge \phi_l$ where each ϕ_i for $1 \leq i \leq l$ is obtained from one of the formulas ψ in Ψ by substituting the variables from ψ by variables from W.
QUESTION: Is Φ satisfiable?

Theorem (BP15)

For all Ψ, Graph-SAT(Ψ) is either in P or NP-Complete. Moreover, the problem of determining Graph-SAT(Ψ) is in P or NP-Complete is decidable.

Gadget - How to Reduce from 3SAT

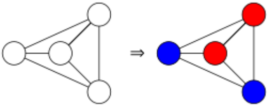
- Variable gadget, clause gadget, wire gadget
- binary logic proof, dual-rail logic proof
- split gadget, NOT gadget, terminator gadget
- ...?

Gadgets are just a tool for (polynomial time) reduction and it varies for the problem uses it. Let's see some example.

Gadget - How to Reduce from 3SAT

Definition (2-Color Perfect Matching)

Given graph G. Is there a 2-coloring of the vertices s.t. every vertex has exactly 1 neighbor of the same color?



Theorem (Sch78)

2-Color Perfect matching is NP-Complete. Moreover, it remains NP-Complete when restricted to planar 3-regular graphs.

Gadget - How to Reduce from 3SAT

Theorem (Sch78)

2-Color Perfect Matching is NP-Complete.

Positive NAE-E3SAT is NP-Complete by Schaefer's Dichotomy Theorem. (Positive NAE-E3SAT: all clauses are form of $x_i \vee x_j \vee x_k$. is it possible to make all clauses have both TRUE and FALSE literals?)

Enough to show: Given φ, can find graph G s.t.
$\varphi \in$ Positive NAE-E3SAT $\Leftrightarrow G$ has 2-Color Perfect Matching

Gadget - How to Reduce from 3SAT

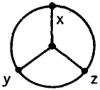


Figure 1(a)

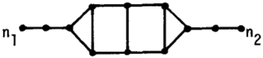


Figure 1(b)

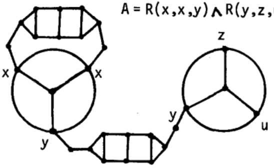


Figure 1(c)

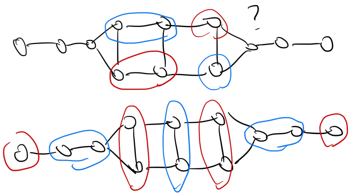
For every clause $X \vee Y \vee Z$ of φ, Figure 1(a) is a clause gadget for it since it guarantees color of X, Y, Z is not all same.

Since one variable can be used many time in φ, we need to separate one variable to multiple vertices in graphs. Figure 1(b) is a variable gadget which guarantees color of vertex n_1 and n_2 to be same.

Figure 1(c) represents a graph corresponds to $A = (X \vee X \vee Y) \wedge (Y \vee Z \vee U)$ composed of two kinds of gadgets. Since building graph by gadgets takes polynomial time,
Positive NAE-E3 $\leq_p$ 2-Color Perfect Matching.

Gadget - How to Reduce from 3SAT

How variable gadget works?



Gadget - How to Reduce from 3SAT

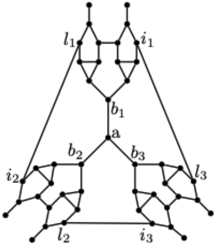


Figure 4: Planar clause gadget K.

- ...Maybe too complicated to show NP-Completeness of planar graph case

Gadget - How to Reduce from 3SAT

$$\begin{array}{ccccc} & S & E & N & D \\ + & M & O & R & E \\ \hline M & O & N & E & Y \end{array}$$

Cryptarithms is NP-Complete, there is a polynomial reduction from Positive 1-IN-3SAT.

Gadget - How to Reduce from 3SAT

Definition (Cryptarithms)

Instance:

- $B, m \in \mathbb{N}$. Let Σ be an alphabet of B letters.
- $x_0, ..., x_{m-1}, y_0, ..., y_{m-1} \in \Sigma$.
- $z_0, ..., z_m \in \Sigma$. The symbol z_m is optional.
- Duplication is allowed within those symbols.

Question:
Is there an injection of Σ into $\{0, 1, ..., B-1\}$ so that satisfies below equation?

$$\begin{array}{ccccc} & & x_{m-1} & \dots & x_0 \\ + & & y_{m-1} & \dots & y_0 \\ \hline & z_m & z_{m-1} & \dots & z_0 \end{array}$$

Gadget - How to Reduce from 3SAT

Theorem

Positive 1-IN-3SAT $\leq_p$ Cryptarithms. Hence Cryptarithms is NP-Complete.

$$\begin{array}{ccc} 0 & p & 0 \\ 0 & p & 0 \\ \hline 1 & q & 0 \end{array}$$

$$\begin{array}{cccccc} 0 & b & c & 0 & a & 0 \\ 0 & b & c & 0 & a & 0 \\ \hline 0 & v & d & 0 & b & 0 \end{array}$$

If the equation's right end is $ApA + ApA = BqA$, then it guarantees that if there is a solution ϕ, $\phi(A) = 0$ and $\phi(B) = 1$. From now, replace A and B to 0 and 1.

Left equation guarantees that $v \equiv 0$ or $v \equiv 1$ modulo 4.

We'll make Cryptarithm such that satisfiability is equivalent and boolean variable FALSE $\Leftrightarrow$ Cryptarithm variable $\equiv 0$ modulo 4, TRUE $\Leftrightarrow$ 1 modulo 4.

Gadget - How to Reduce from 3SAT

Theorem

Positive 1-IN-3SAT $\leq_p$ Cryptarithms. Hence Cryptarithms is NP-Complete.

$$\begin{array}{ccccccccccc} 0 & I & 0 & x & 0 & 1 & 0 & b & 0 & a & 0 \\ 0 & z & 0 & y & 0 & c & 0 & b & 0 & a & 0 \\ \hline 0 & d & 0 & I & 0 & d & 0 & c & 0 & b & 0 \end{array}$$

Above equation is a clause gadget of $x \vee y \vee z$. Since the problem is Positive 1-IN-3SAT, we need to guarantee $x + y + z \equiv 1$ modulo 4. We can see $d = 4a + 1 \equiv 1$ modulo 4 and $x + y + z = I + z = d$, Therefore $x + y + z \equiv 1$ modulo 4.

Exercise

Write formal proof of Positive 1-IN-3SAT $\leq_p$ Cryptarithms.

More problems in the book

There are more proofs of NP-Completeness of problems I did not include in this slides.

- Grid Coloring
- Pushing 1x1 block
- PushPush
- Super Mario Brothers

I'll recommend read this if you are interested in reduction of real games.

Thank you!