
15-455: UNDERGRADUATE COMPLEXITY THEORY, SPRING 2017, RYAN O'DONNELL

A PREPRINT

Yang Li

Faculty of Information Technology
Monash University
Clayton, VIC, Australia
yang.kelvinli@monash.edu

March 22, 2019

Lecture 6 SAT

formula-sat, cnf-sat, k-sat, 3-sat,

1 CIRCUIT-SAT

Definition 1.1. Boolean circuit C is a DAG, in which nodes are called gates, and edges are called wires... usually has n inputs and 1 output... gates are and \wedge , or \vee , negation \neg with fan-in 2, 2, 1 respectively and arbitrary number of fan-out.

A circuit C computes a Boolean function which maps n bits strings into one bit $\{0, 1\}^n \rightarrow \{0, 1\}$.

You can think of circuit as a computational model.

One circuit can only operations on inputs of a fixed length, because number of sources in the DAG is fixed. This is one of the main reason we don't use it as the first choice of our computational model.

When the input to algorithm is a circuit $\langle C \rangle$, we measure run time in terms of $n :=$ number of input gates and $m :=$ the number of total gates. These parameters are polynomially related to the input size, so if we only care if the algorithm can run in P time or not, whether the run time is measured by the size of the input or n, m doesn't matter. $n \leq m \leq |\langle C \rangle| \leq O(m \log m)$.

Definition 1.2 (circuit-eval). given $\langle C, x = \{0, 1\}^n \rangle$, what is $C(x)$? This can also be written into a language $\{\langle C, x \rangle : C(x) = 1\}$.

This can be computed in P time. It's the same as checking an answer of a circuit. Roughly speaking, you can do it in $O(m)$.

Definition 1.3 (circuit-sat). given $\langle C \rangle$, is there an x s.t. $C(x) = 1$? That is, $\{\langle C \rangle : \exists x \text{ s.t. } C(x) = 1\}$.

Brute force, try all 2^n xs, find the one that gives the true answer. This is probably the best we can do. We think that it can be done in P iff $P = NP$.

2 FORMULA-SAT

Definition 2.1. A formula is a circuit where all the fan-outs are 1.

eg, $(x_1 \wedge x_2) \wedge (x_2 \vee x_3) \vee (\neg(x_1 \wedge x_2))$. A circuit can be represented by a formula.

Formula-sat is a special case of a circuit-sat. Any formula is an example of a circuit. It's potentially easier, but we still don't have a smarter way of computing it.

3 CNF-SAT

Definition 3.1. Special case of formula-sat, where the formula is a conjunction of clauses, and a clause is a disjunction of literals, a literal is a variable or its negation.

For a cnf problem, n is the number of variables, and m is the number of clauses. It's a potential even easier problem. It is in EXP time $O(2^nm)$. It is NP-complete.

Definition 3.2. k-sat is a special case of cnf-sat, where all clauses have at most k literals.

it's potential easier than cnf. A popular example is 3-sat. WLOG, in 3-sat, $m \leq O(n^3)$, this is over counting a lit, but it's ok. We know it's NP-complete. It's in time 1.34^n .

4-sat is in 1.5^n , 5-sat in 1.6^n , 6-sat 1.667^n , ..., k-sat in time $(2 - 2/k)^n \text{poly}(n)$. 2-sat is in polynomial time.

Theorem 3.1. 2-sat is in P.

e.g. input: $(x_1 \vee x_2) \wedge (x_1 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2) \wedge (x_2 \vee x_3) \wedge (\bar{x}_3)$

Proof. observation: $(x_1 \vee x_2) = (\bar{x}_1 \rightarrow x_2) = (\bar{x}_2 \rightarrow x_1)$ because of the truth table for disjunction $(x_1 \vee x_2)$ is the same as the truth table for the implication $(\bar{x}_1 \rightarrow x_2)$. See https://en.wikipedia.org/wiki/Truth_table

rewrite the above expression using the \rightarrow formulas, using both directions. ie., $(\bar{x}_1 \rightarrow x_2) \wedge (\bar{x}_2 \rightarrow x_1) \wedge \dots$

draw a directed graph with $2n$ vertices, one for each literal and $2m$ directed edges, one for each implication.

key observation of the directed graph: if $l_i \rightarrow l_j$ is an edge, l is a literal, then there is also an edge $\bar{l}_j \rightarrow \bar{l}_i$.

There is $\bar{x}_2 \rightarrow x_1 \rightarrow x_2$, i.e., there is a path $\bar{x}_2 \rightarrow x_2$. In any satisfiable assignment, x_2 better be TRUE.

Suppose exists a path $x_2 \rightarrow \bar{x}_2$, then x_2 better be FALSE.

If from some i , there exists a path $x_i \rightarrow \bar{x}_i$ and $\bar{x}_i \rightarrow x_i$, then the instance is unsatisfiable. call them contradictory path.

In $\text{poly}(n)$ time, can check whether exists contradictory paths.

It remains to show that if there is no such contradictory path, then it is satisfiable.

assume for all i , we don't have contradictory path. □

References