Lec27-using-3SAT

Monday, November 18, 2024 9:48 AM

HT Chapter 8

Cook-Levin Theorem (1971): The SAT problem is NP-complete (Boolean Satisfiability)

Both Boolean (each true or false) and Integer (can be assigned any value)

SIGACT (Special Interest Group on Algorithms and Computation Theory)

Any problem in NP

↓

SAT

↓

3SAT

↓ ↓ ↓

Independent Set | Hamiltonian Cycle | Graph Coloring | 3-dimensional matching

↓ | ↓ | |

Vertex Cover | Hamiltonian Path | |

↓ | ↓ | |

Set Cover | Traveling Salesman | |

Boolean Formula

Variables: $x_1, x_2, x_3, \dots \in \{T, F\}$ (Boolean)

Terms: $t_1, t_2, t_3, \dots, t_n$, where $t_i = x_i$ or $\overline{x_i}$ (not x_i)

OR clause: $t_{i_1} \vee t_{i_2} \vee t_{i_3} \vee \dots \vee t_{i_k}$ (OR of literals (terms) that are negations)

CNF: AND of ORs. Let $C_1, \dots, C_m$ be clauses.

(conjunctive normal form)

$C_1 \wedge C_2 \wedge \dots \wedge C_m$ is a Boolean formula in CNF.

Ex $(x_1 \vee \overline{x_1}) \wedge (\overline{x_2} \vee \overline{x_3}) \wedge (x_1 \vee x_3)$

Ex $(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_1} \vee x_3)$

Def: A truth assignment is a choice of T/F for each variable i.e. a function $v: X \to \{T, F\}$.

Ex $x_1 = T$, $x_2 = T$, $x_3 = F$

Ex $x_1 = T$, $(x_1 = F, x_3 = F)$

Def: A truth assignment is a satisfying assignment for a Boolean formula if it make the formula true.

(for CNF, if every clause is set to True)

Problem (SAT): Given a set of OR-clauses $C_1, \dots, C_m$ over variables $X = \{x_1, \dots, x_n\}$, is there a satisfying assignment?

Problem (3SAT): Given a set of OR-clauses $C_1, \dots, C_m$, each of length 3, over variables $X = \{x_1, \dots, x_n\}$, is there a satisfying assignment?

} reducible decision problems

Reducing SAT to 3SAT

(to show 3SAT is NP-hard)

Given SAT instance $C = \{C_1, \dots, C_m\}$. Suppose $|C_i| \neq 3$.

$C_i = (t_1 \vee t_2 \vee t_3 \vee \dots \vee t_k)$ ⟵ Term: If any one of variables t_j is T.

Replace clause with set of clauses with new Y_i variables

$(t_1 \vee t_2 \vee y_1)$

$(\overline{y_1} \vee t_3 \vee y_2)$

$(\overline{y_2} \vee t_4 \vee y_3)$

$\vdots$

$(\overline{y_{k-3}} \vee t_{k-1} \vee t_k)$

If all clause are true, then at least one $y_i = T$. Suppose not. Then $y_1 = T$ $\Rightarrow y_2 = T$, $y_3 = T$, ... but then last clause is F. $\Rightarrow$ all clause = T implies original clause was T.

If $t_1 = T$, can set $y_1 = T$, $y_2 = T$, ... $y_{k-3} = T$ $\Rightarrow$ $(\overline{y_{k-3}} \vee t_{k-1} \vee t_k)$ is T.

else

$(\overline{y_1} \vee F \vee y_2)$

$(\overline{y_2} \vee F \vee y_3)$

$\vdots$

$(\overline{y_{k-3}} \vee t_{k-1} \vee t_k)$

must have $y_{k-3} = F$ $\Rightarrow$ $(\overline{y_{k-3}} \vee t_{k-1} \vee t_k)$ is T.

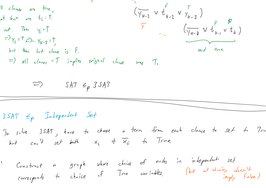
$\Rightarrow$ SAT $\leq_p$ 3SAT

Thm: 3SAT $\leq_p$ Independent Set

proof: To solve 3SAT, have to choose one term from each clause to set to True but can't set both x_i & $\overline{x_i}$ to True

Strategy: Construct a graph where choice of nodes is independent set corresponds to choice of True variables. (set of nodes = assign True)

Ex $(x_1 \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3)$



triangle gadget means that given k clauses, will find set of at most size k

negation gadget lets x_i with all $\overline{x_i}$'s in your couple can get both x_i & $\overline{x_i}$ simultaneously to True.

If formula is satisfiable, at least one term (true) in each clause. Let S be a set of one true term from each clause. $|S| = k$ and no two nodes in S are connected by an edge (otherwise, would get $x_i = T = \overline{x_i}$)

If graph has independent set $|S| = k$, must have one node from each triangle gadget. Set them terms to true in original 3SAT formula, gives a solution to 3SAT.

General strategy for NP-complete proofs: (cont)

1. Show $X \in NP$ by showing that there is a certificate that is efficiently checked
2. Look at known NP-complete problems. Choose a Y that seems "similar" to X
3. Show that $Y \leq_p X$

   A) Let I_Y be any instance of Y

   B) Construct instance I_X in poly time s.t.
   - If I_Y is Yes-instance, then I_X is yes-instance
   - If I_X is Yes-instance, then I_Y is yes-instance

So if you can solve X, then you can solve Y. (but not vice versa, since we are converting Y instance to X instance)

Problem (Hamiltonian Cycle): Given directed graph G, is there a Cycle that visits every vertex exactly once. (and then returns to start)

known as Hamiltonian Cycle

Theorem: Hamiltonian Cycle is NP-complete

proof: HamCycle $\in$ NP, because given a cycle it straightforward to check all nodes visited & all edges used.

Want to show 3SAT $\leq_p$ HamCycle

Need to create 3SAT instance I_H HamCycle instance.

Variables $x_1, \dots, x_n$ } constructed "gadgets" representing sets of Clauses $C_1, \dots, C_m$ } hook them up in graph

Show this graph has Ham cycle IFF formula is satisfiable

Variable gadget



Note that once you choose a direction, you can only go in that direction.

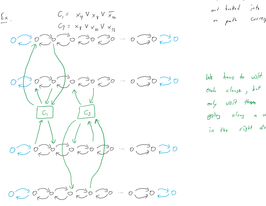
Left ⟵ correspond to setting $x_i = T$

Right ⟶ correspond to setting $x_i = F$.

Clause gadget: Add a single node for $C_j = t_1 \vee t_2 \vee t_3$ & hook into the variables correspondingly to t_1, t_2, t_3, with direction specified by if $t_i = x_i$ or $\overline{x_i}$ we hooked into the x_i or path correspondingly to j.

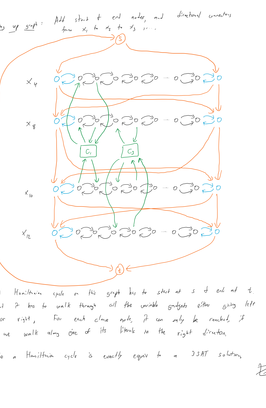
Ex $C_1 = x_1 \vee x_2 \vee \overline{x_3}$

$C_2 = x_1 \vee x_2 \vee x_3$



We have to visit each clause, but can only visit them while going along a variable in the right direction

Connecting up 3SAT: Add start & end nodes and directional connectors from x_i to x_j as $i < j$.



If Hamiltonian cycle in this graph has to start at s & end at t. But it has to walk through all the variable gadgets either going left or right, for each clause node. It can only be reached if we walk along one of its blocks in the right direction.

So a Hamiltonian cycle is exactly equiv to a 3SAT solution.