http://www.cs.unh.edu/~ruml/cs758

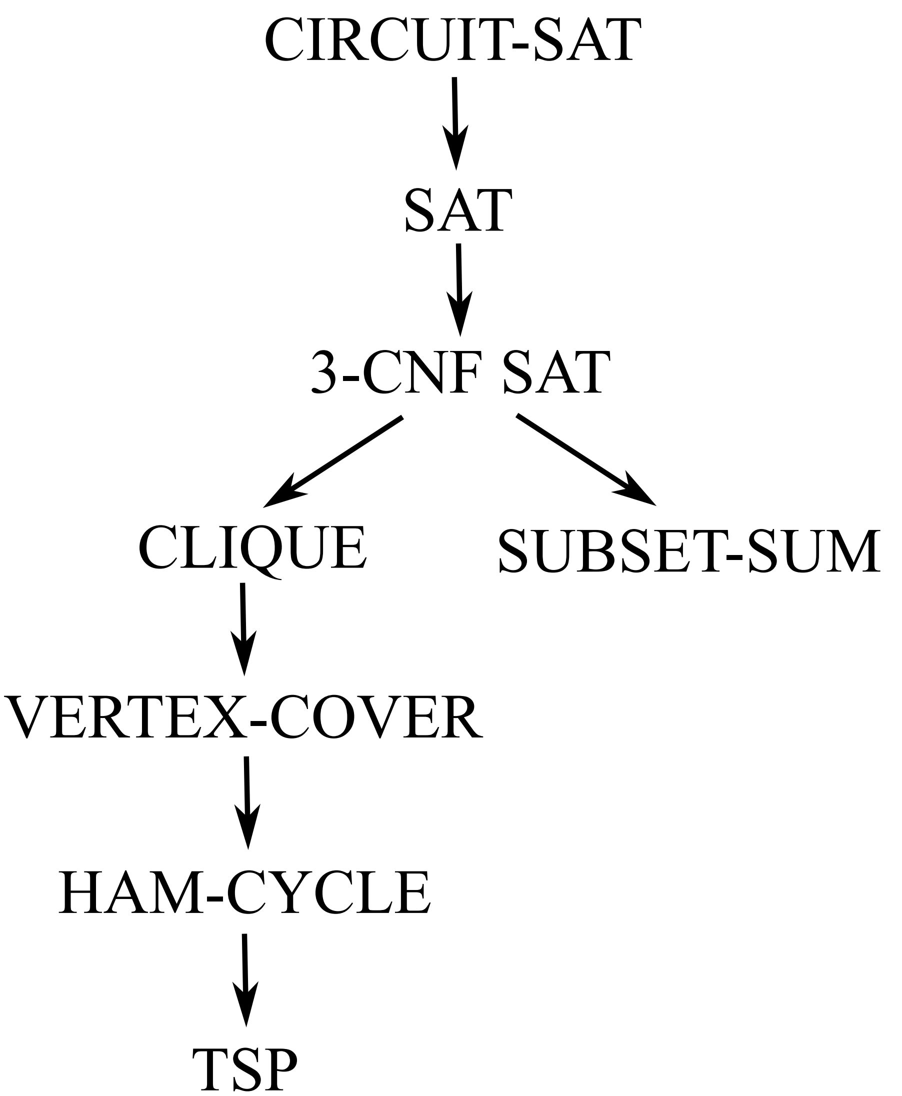
Framework for an NP-Completeness Proof

To prove some problem A is NP-complete:

1. Prove $A \in NP$
2. Prove A is NP-hard.
   (a) Pick a known NP-complete problem B
   (b) Design a reduction that translates instances of B into equivalent instances of A
      i. Show that translated A version is accepted if and only if the original B version should be accepted.
      ii. Prove that the reduction runs in polynomial time.

Reductions to Graph Problems

Reductions



CIRCUIT-SAT
SAT
3-CNF SAT
CLIQUE
SUBSET-SUM
VERTEX-COVER
HAM-CYCLE
TSP

Clique

Given graph G and integer $k > 1$, does G have clique of size k?

CLIQUE $\in$ NP: given clique, test connectivity (k^2 time).

CLIQUE is NP-Hard: Reduction from 3-CNF SAT! Formula ϕ with k clauses will be SAT iff graph G has a k clique.

For clause r like $(l_1^r \vee l_2^r \vee l_3^r)$, add vertices v_1^r, v_2^r, and v_3^r to G. Add edge from v_i^r to v_j^s iff $r \neq s$ and $l_i^r \neq \neg l_j^s$.

SAT $\Rightarrow$ clique: If ϕ SAT, at least one literal in each clause is true. These form a clique in G because they cannot conflict. Clique $\Rightarrow$ SAT: If k clique, make corresponding literals true. Will satisfy all k clauses without conflicts.

Example: $(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_3)$

Vertex Cover

Given graph G and integer $k > 0$, does G have a vertex cover of size k?

VERTEX-COVER $\in$ NP: given cover, check size and that each edge is covered.

VERTEX-COVER is NP-Hard: Reduction from CLIQUE. Form graph complement $\overline{G}$, which has edge (u, v) for $v \neq u$ iff original does not. Claim: G has k clique iff $\overline{G}$ has $|V| - k$ cover.

Cover $\Rightarrow$ clique: All edges in $\overline{E}$ have at least one endpoint in $Cover$. All pairs (u, v) with both u and $v \notin Cover$ therefore have edge $\in E$. So $V - Cover$ is a clique of size k.

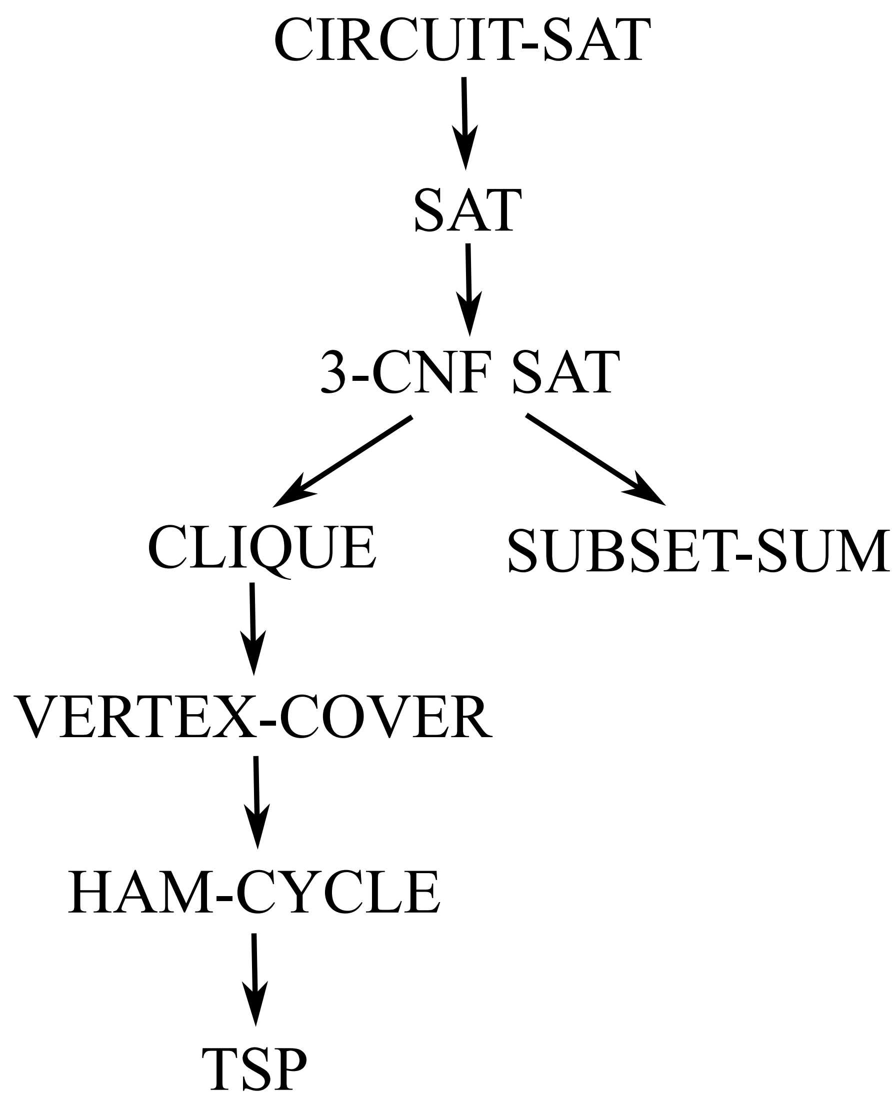
Clique $\Rightarrow$ cover: Any edge $(u, v) \in \overline{E}$ implies $\notin E$ implies u or v not in $Clique$. This implies u or v remains in $V - Clique$ and hence it covers that edge. Size of $V - Clique$ is $|V| - k$.

Break

- asst 12
- Wildcard vote!

Reduction to a Numeric Problem

Reductions



CIRCUIT-SAT
SAT
3-CNF SAT
CLIQUE
SUBSET-SUM
VERTEX-COVER
HAM-CYCLE
TSP

Subset Sum

Given finite set of positive integers, is there a subset that sums to t?

SUBSET-SUM $\in$ NP: given subset, compute sum.

SUBSET-SUM is NP-Hard: Reduction from 3-CNF SAT. Make numbers and the target sum from the formula. For n variables and k clauses, each number will have $n + k$ digits. We ensure no carrying by using base 10 and at most a sum of 6 in each column.

[see upcoming slide for how to make numbers and target]

Polynomial time to construct and equivalent to satisfiability.

Example Formula

$$
\begin{aligned}
C_1 : &\quad (x_1 \vee \neg x_2 \vee \neg x_3) \wedge \\
C_2 : &\quad (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge \\
C_3 : &\quad (\neg x_1 \vee \neg x_2 \vee x_3) \wedge \\
C_4 : &\quad (x_1 \vee x_2 \vee x_3)
\end{aligned}
$$

Subset Sum

Two kinds of numbers:

- Two numbers for each variable, representing positive/negative literals. (These are the 'important' ones!) 1 in the variable's column, and 1 for clauses where that literal appears.
- Clause numbers just allow slop for 1, 2 or 3 true literals per clause.

Target is 1 for each variable and 4 for each clause. Therefore, it requires exactly one form of each variable and at least one true literal in each clause (plus one or both 'slop numbers').

Sum $\Rightarrow$ SAT: read off assignment. Target ensures consistency and variable numbers ensure satisfiability.

SAT $\Rightarrow$ sum: construct sum, choosing slop variables last.

Resulting Set

	x_1	x_2	x_3	C_1	C_2	C_3	C_4
$v_1 =$	1	0	0	1	0	0	1
$v'_1 =$	1	0	0	0	1	1	0
$v_2 =$	0	1	0	0	0	0	1
$v'_2 =$	0	1	0	1	1	1	0
$v_3 =$	0	0	1	0	0	1	1
$v'_3 =$	0	0	1	1	1	0	0
$s_1 =$	0	0	0	1	0	0	0
$s'_1 =$	0	0	0	2	0	0	0
$s_2 =$	0	0	0	0	1	0	0
$s'_2 =$	0	0	0	0	2	0	0
$s_3 =$	0	0	0	0	0	1	0
$s'_3 =$	0	0	0	0	0	2	0
$s_4 =$	0	0	0	0	0	0	1
$s'_4 =$	0	0	0	0	0	0	2
$t =$	1	1	1	4	4	4	4

	x_1	x_2	x_3	C_1	C_2	C_3	C_4
$v_1 =$	1	0	0	1	0	0	1
$v'_1 =$	1	0	0	0	1	1	0
$v_2 =$	0	1	0	0	0	0	1
$v'_2 =$	0	1	0	1	1	1	0
$v_3 =$	0	0	1	0	0	1	1
$v'_3 =$	0	0	1	1	1	0	0
$s_1 =$	0	0	0	1	0	0	0
$s'_1 =$	0	0	0	2	0	0	0
$s_2 =$	0	0	0	0	1	0	0
$s'_2 =$	0	0	0	0	2	0	0
$s_3 =$	0	0	0	0	0	1	0
$s'_3 =$	0	0	0	0	0	2	0
$s_4 =$	0	0	0	0	0	0	1
$s'_4 =$	0	0	0	0	0	0	2
$t =$	1	1	1	4	4	4	4

EOLQs

For example:

- What's still confusing?
- What question didn't you get to ask today?
- What would you like to hear more about?

Please write down your most pressing question about algorithms and put it in the box on your way out.
Thanks!