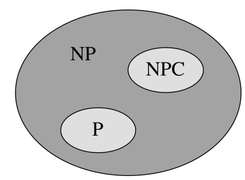
CS 140 Algorithms – Fall 2013

Prof. Jim Boerkoel

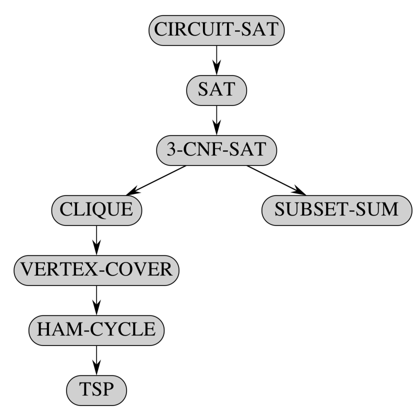
Lecture 22. NP-Completeness Reductions

1 Important Terminology and Concepts.

- Tractability:
- Intractability:
- NP:



- P:
- NP Hard:
- NP Complete:
- Optimization vs. Decision (e.g., MST vs. MST-Decision):
- Reduction Function: ($\leq_P$):
- Cook-Levin Theorem: *Every decision problem in NP can be (quickly) converted into a corresponding 3-SAT decision problem.*



- Proving NP-Completness:
- Cool resource! `https://complexityzoo.uwaterloo.ca`

2 Subset-sum

2.1 Problem Description

In the ***subset-sum problem***, we are given a finite set S of positive integers and an integer ***target*** $t > 0$. We ask whether there exists a sebset $S' \subseteq S$ whose elements sum to t. Formally:
SUBSET-SUM $= \{\langle S, t \rangle :$ there exists a subset $S' \subseteq S$ such that $t = \sum_{s \in S'} s\}$.

2.2 Example

If $S = \{1, 2, 7, 14, 49, 98, 343, 686, 2409, 2793, 16808, 17206, 117705, 117993\}$ and
$t = 138457$, then the subset
$S' = \{1, 2, 7, 98, 343, 686, 2409, 17206, 117705\}$ is a solution.

2.3 Proof of NP-Completeness

Theorem 34.15 The subset-sum problem is NP-complete.

2.3.1 Subset sum is in NP

A) Subset sum Verification Algorithm:

B) Subset sum Verification Runtime:

2.3.2 Subset sum is NP-hard

We now show that the subset-sum problem is NP-hard by showing that 3-CNF-SAT $\leq_P$ SUBSET-SUM.

- Given: a 3-CNF formula ϕ over variables $x_1, x_2, ..., x_n$ with clauses $C_1, C_2, ..., C_k$ each containing 3 distinct literals.
- Assumption 1 (WLOG): no clause contains both a variable and its negation (inherently satified).
- Assumption 2 (WLOG): each variable appears in at least one clause (otherwise we wouldn't care what value is assigned to the superfluous variables).

Our reduction algorithm constructs an instance $\langle S, t\rangle$ of the subset-sum problem such that ϕ is satisfiable if and only if there exists a subset of S whose sum is exactly t.

A) Reduction Algorithm:

- Our reduction will create two numbers in set S for each variable and for each clause, for a total of 2(n+k) numbers.
- Each number is base 10, where each number contains $n+k$ digits and each digit corresponds to either one variable or one clause.
- The target t has a 1 in each digit labeled by a variable and a 4 in each digit labeled by a clause.
- For each variable x_i, set S contains two integers v_i and v_i', representing x_i and $\neg x_i$ respectively, that contain a 1 in the digit corresponding to x_i, a 1 in any digit that corresponds to a clause containing x_i or $\neg x_i$, respectively, and 0's elsewhere. By our assumptions, all values in S are unique.
- For each clause C_j, set S contains two 'slack' integers s_j and s_j', each of which have all 0's in all digits other than the one labeled by C_j; for s_j there is a 1 in this digit, and for s_j' there is a 2 in this digit.

B) Reduction Runtime:

		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

Figure 1: The reduction of 3-CNF-SAT to Subset-Sum. The formula in 3-CNF is $\phi = C_1 \wedge C_2 \wedge C_3 \wedge C_4$, where $C_1 = (x_1 \vee \neg x_2 \vee \neg x_3), C_2 = (\neg x_1 \vee \neg x_2 \vee \neg x_3), C_3 = (\neg x_1 \vee \neg x_2 \vee x_3)$, and $C_4 = (x_1 \vee x_2 \vee x_3)$

C) The 3-CNF formula ϕ is satisfiable if and only if there exists a subset $S' \subseteq S$ whose sum is t.

- IFF $\Rightarrow$ (yes$\rightarrow$yes):

- IFF $\Leftarrow$ (no$\rightarrow$no):

Conclusion: