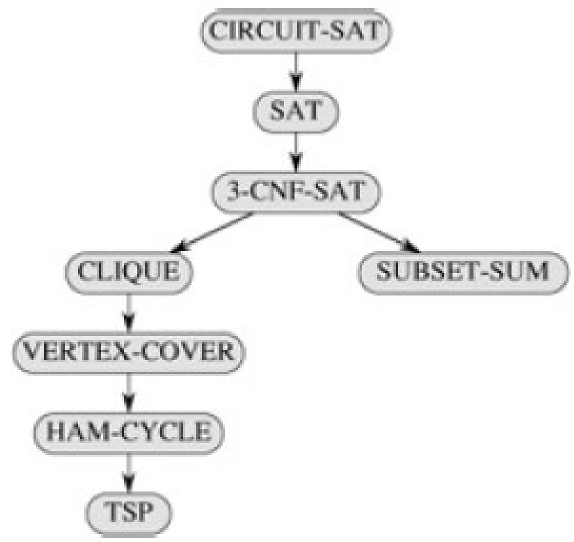
Structure of Proof



Subset Sum $\leq_R$ 0/1 Knapsack Problem

2.1 subset sum problem

Given a set S of n positive integers, and a positive value SUM, does there exist a subset S' of S for which sum of all elements $\sum S'_i = SUM$?

Input : $S = \{3, 2, 7, 1\}$, $SUM = 6$
Output: $True$, subset is $S' = \{3, 2, 1\}$

2.2 0/1 Knapsack Problem

Given a set I of n items, with corresponding profit values P, weight values W, Capacity C and max-profit V; does there exists a subset I' of items I for which total weight $\sum W'_i \leq C$ and total profit $\sum P'_i \geq V$?

Input : $I = \{i_1, i_2, i_3, i_4\}$, $P = \{4, 2, 6, 8\}$, $W = \{2, 3, 1, 2\}$, $C = 5$, $V = 14$
Output : $True$, $I' = \{i_3, i_4\}$

3 Reduction Procedure

a. **Take an instance of subset sum :** $S = \{3, 2, 7, 1\}$, $SUM = 6$

b. **Define a procedure to create an instance of 0/1 knapsack using the instance of subset sum:**

$|I| = n$, where n = no. of elements in S.

$V = SUM$

$C = SUM$

$P = S$

$W = S$

so, $I = \{i_1, i_2, i_3, i_4\}$, $P = \{3, 2, 7, 1\}$, $W = \{3, 2, 7, 1\}$, $V = 6$, $C = 6$

c. **Solve 0/1 knapsack problem:** To prove the reduction we assume that there exist a polynomial time algorithm to solve 0/1 knapsack problem. But here to write the program and to understand the procedure, we shall use dynamic programming algorithm to solve 0/1 knapsack. The solution for the instance given in step b is "True" i.e. there is a subset of items for which total profit = 6, and total weight = 6.

d. **Find solution of subset sum from the solution of 0/1 knapsack :** $Solution(subsetsum) = solution(0/1knapsack)$

$Solution(subsetsum) = True$

e. **Show that b and d takes polynomial time :** Step b and d are assignments of finite size, hence both can be done in constant time.

Reduction Procedure

Algorithm 1 Algo_subsetsum ($S = \{s_1, s_2, ...s_n\}, SUM$)

$P \leftarrow S$
$W \leftarrow S$
$C \leftarrow SUM$
$V \leftarrow SUM$
if Algo_dynamic_knapsack(P, W, C, V) == "TRUE" **then**
  RETURN "TRUE"
else
  RETURN "FALSE"
end if

Reduction Procedure

Algorithm 2 Algo_dynamic_knapsack (P, W, C, V)

```
Create M[0...N][0...C] //NXC dimension matrix to store solutions of sub
problems
while w from 0 to C do
  M[0, w] ← 0
end while
while i from 1 to n do
  M[i, 0] = 0
end while
while i from 1 to n do
  while w from 0 to C do
    if W_i ≤ w then
      if P_i + M[i − 1, w − W_i] > M[i − 1, w] then
        M[i, w] ← P_i + M[i − 1, w − W_i]
      else
        M[i, w] ← M[i − 1, w]
      end if
    else
      M[i, w] ← M[i − 1, w]
    end if
  end while
end while
if M[N, W] ≥ V then
  RETURN "TRUE"
else
  RETURN "FALSE"
end if
```

Exercise

Run the algorithm for the following instances of the subset sum problem,

(i) $S = \{4, 3, 6, 8, 5, 9\}$, $SUM = 23$

(ii) $S = 3, 5, 7, 9, 11$, $SUM = 13$