CS F364
Design & Analysis of Algorithms

ALGORITHMS - COMPLEXITY

Complexity Classes > NP-Completeness Via Reductions

- Reduction Techniques:
  - Restriction
  - Example: 0,1 Knapsack

Problem: 0,1 Knapsack

- KNAPSACK:
  - Given a set S of items with weights { W1, W2, …, Wn} and values { P1, P2, …, Pn} , a weight bound B, a value bound V,
    - find whether there is a subset T of S, such that
      - the sum of the weights of elements in T is <= B, and that
      - the sum of values of elements in T is >=V.

KNAPSACK IS $\mathbb{NP}$-HARD

- KNAPSACK is $\mathbb{NP}$-hard
  - SUBSET-SUM $\preceq$ KNAPSACK
  - Proof:
    - SUBSET-SUM is a special case of KNAPSACK
      - with Wi = Pi for all i, and V = B
- Proof By Restriction:
  - Show that the known problem is a special case of the target problem.

Reduction Technique: Restriction

- Observe that the proof in the previous example:
  - is an instance of ***Reduction by Restriction***:
    - i.e. we show that the known $\mathbb{NP}$ -hard problem is a special case of the target problem.
  - Why does this reduction work?
    - Will it work the other way round?
- Exercise:
  - Observe the way this technique was used in reducing HAM-CIRCUIT to TSP