NP-Complete Problems

CMSC 451, Summer 2009

General NP-Complete Proof Outline

We want to show that UNKNOWN-PROBLEM is NP-Complete.

1) Show that UNKNOWN-PROBLEM is in NP.
2) Show that NP-COMPLETE-PROBLEM ≤ p UNKNOWN-PROBLEM
   a) Pick the specific problem to be the NP-COMPLETE-PROBLEM. Consider problems that are of the same type - packing, covering, sequencing, etc.
   b) Give a description of a translation between problem instances I of type NP-COMPLETE-PROBLEM and instances I' of type UNKNOWN-PROBLEM.
   c) Prove that the solution to I exists if and only if the solution to I' exists. This should hold for all instances of type NP-COMPLETE-PROBLEM and all instances described by the translation from part b) of type UNKNOWN-PROBLEM. In other words, it does not need to hold for all instances of type UNKNOWN-PROBLEM.
### 8.6 Partitioning Problems

**Basic genres.**
- Packing problems: SET-PACKING, INDEPENDENT SET.
- Covering problems: SET-COVER, VERTEX-COVER.
- Constraint satisfaction problems: SAT, 3-SAT.
- Sequencing problems: HAMILTONIAN-CYCLE, TSP.
- **Partitioning problems**: 3D-MATCHING, 3-COLOR.
- Numerical problems: SUBSET-SUM, KNAPSACK.

---

#### 3-Dimensional Matching

**3D-MATCHING.** Given n instructors, n courses, and n times, and a list of the possible courses and times each instructor is willing to teach, is it possible to make an assignment so that all courses are taught at different times?

<table>
<thead>
<tr>
<th>Instructor</th>
<th>Course</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Friedler</td>
<td>CMSC 451</td>
<td>MW 11-12:20</td>
</tr>
<tr>
<td>Friedler</td>
<td>CMSC 451</td>
<td>TTh 11-12:20</td>
</tr>
<tr>
<td>Friedler</td>
<td>CMSC 330</td>
<td>TTh 11-12:20</td>
</tr>
<tr>
<td>Friedler</td>
<td>CMSC 351</td>
<td>TTh 11-12:20</td>
</tr>
<tr>
<td>Tardos</td>
<td>CMSC 250</td>
<td>TTh 3-4:20</td>
</tr>
<tr>
<td>Tardos</td>
<td>CMSC 451</td>
<td>TTh 11-12:20</td>
</tr>
<tr>
<td>Tardos</td>
<td>CMSC 451</td>
<td>TTh 3-4:20</td>
</tr>
<tr>
<td>Kleinberg</td>
<td>CMSC 330</td>
<td>TTh 3-4:20</td>
</tr>
<tr>
<td>Kleinberg</td>
<td>CMSC 330</td>
<td>MW 11-12:20</td>
</tr>
<tr>
<td>Kleinberg</td>
<td>CMSC 451</td>
<td>MW 11-12:20</td>
</tr>
</tbody>
</table>
Recall Bipartite Matching

Bipartite matching.
- Input: undirected, bipartite graph $G = (L \cup R, E)$.
- $M \subseteq E$ is a matching if each node appears in at most one edge in $M$.
- Max matching: find a max cardinality matching.
- Decision problem: Does there exist a matching of cardinality $\geq k$?

Rephrasing:
- We are given two sets $L$ and $R$ and a set of pairs drawn from $L \times R$.
- Does there exist a set of pairs of size $\geq k$ such that each element of $L \cup R$ appears in exactly one pair?

3-Dimensional Matching

3D-MATCHING. Given disjoint sets $X$, $Y$, and $Z$, each of size $n$ and a set $T \subseteq X \times Y \times Z$ of triples, does there exist a set of $n$ triples in $T$ such that each element of $X \cup Y \cup Z$ is in exactly one of these triples?

Claim. 3-SAT $\leq_p$ 3D-MATCHING.

Pf. Given an instance $\Phi$ of 3-SAT, we construct an instance of 3D-matching that has a perfect matching iff $\Phi$ is satisfiable.

See the book for the proof.
Group Practice Problems

Note: We are NOT proving that 3D-MATCHING is NP-Complete here.

3D-MATCHING. Given disjoint sets X, Y, and Z, each of size n and a set T ⊆ X × Y × Z of triples, does there exist a set of n triples in T such that each element of X ∪ Y ∪ Z is in exactly one of these triples?

VERTEX COVER: Given a graph G = (V, E) and an integer k, is there a subset of vertices S ⊆ V such that |S| ≤ k, and for each edge, at least one of its endpoints is in S?

1 Prove that 3D-MATCHING ≤p VERTEX COVER.

SET PACKING: Given a set U of n elements, a collection S₁, S₂, . . . , Sₘ of subsets of U, and an integer k, does there exist a collection of at least k of these sets with the property that no two of them intersect?

2 Prove that 3D-MATCHING ≤p SET PACKING.

8.7 Graph Coloring

Basic genres.

- Packing problems: SET-PACKING, INDEPENDENT SET.
- Covering problems: SET-COVER, VERTEX-COVER.
- Constraint satisfaction problems: SAT, 3-SAT.
- Sequencing problems: HAMILTONIAN-CYCLE, TSP.
- Partitioning problems: 3D-MATCHING, 3-COLOR.
- Numerical problems: SUBSET-SUM, KNAPSACK.
3-Colorability

**3-COLOR:** Given an undirected graph $G$ does there exist a way to color the nodes red, green, and blue so that no adjacent nodes have the same color?

**COLOR:** Given an undirected graph $G$ and an integer $k$ does there exist a way to color the nodes using $k$ colors so that no adjacent nodes have the same color? (NP-Complete for $k > 2$)

Register Allocation

**Register allocation.** Assign program variables to machine register so that no more than $k$ registers are used and no two program variables that are needed at the same time are assigned to the same register.

**Interference graph.** Nodes are program variables names, edge between $u$ and $v$ if there exists an operation where both $u$ and $v$ are "live" at the same time.

**Observation.** [Chaitin 1982] Can solve register allocation problem iff interference graph is $k$-colorable.

**Fact.** $3$-COLOR $\leq_p k$-REGISTER-ALLOCATION for any constant $k \geq 3$. 
3-Colorability

Claim. 3-SAT \leq_p 3-COLOR.

Pf. Given 3-SAT instance \( \Phi \), we construct an instance of 3-COLOR that is 3-colorable iff \( \Phi \) is satisfiable.

See book for the proof.

Practice Problem

3D-MATCHING. Given disjoint sets \( X, Y, \) and \( Z \), each of size \( n \) and a set \( T \subseteq X \times Y \times Z \) of triples, does there exist a set of \( n \) triples in \( T \) such that each element of \( X \cup Y \cup Z \) is in exactly one of these triples?

3-COLOR: Given an undirected graph \( G \) does there exist a way to color the nodes red, green, and blue so that no adjacent nodes have the same color?

Prove 3D-MATCHING \( \leq_p \) 3-COLOR.
8.8 Numerical Problems

Basic genres.
- Packing problems: SET-PACKING, INDEPENDENT SET.
- Covering problems: SET-COVER, VERTEX-COVER.
- Constraint satisfaction problems: SAT, 3-SAT.
- Sequencing problems: HAMILTONIAN-CYCLE, TSP.
- Partitioning problems: 3-COLOR, 3D-MATCHING.
- Numerical problems: SUBSET-SUM, KNAPSACK.

Subset Sum

**SUBSET-SUM.** Given natural numbers \(w_1, \ldots, w_n\) and an integer \(W\), is there a subset that adds up to exactly \(W\)?

**Ex:** \(\{1, 4, 16, 64, 256, 1040, 1041, 1093, 1284, 1344\}\), \(W = 3754\).
Yes. \(1 + 16 + 64 + 256 + 1040 + 1093 + 1284 = 3754\).

**Remark.** With arithmetic problems, input integers are encoded in binary. Polynomial reduction must be polynomial in binary encoding.

**Claim.** \(3\text{-SAT} \leq_p \text{SUBSET-SUM}\).
**Pf.** Given an instance \(\Phi\) of 3-SAT, we construct an instance of SUBSET-SUM that has solution iff \(\Phi\) is satisfiable.
Scheduling With Release Times

**SCHEDULE-RELEASE-TIMES.** Given a set of n jobs with processing time \( t_i \), release time \( r_i \), and deadline \( d_i \), is it possible to schedule all jobs on a single machine such that job \( i \) is processed with a contiguous slot of \( t_i \) time units in the interval \([r_i, d_i]\)?

**Claim.** \( \text{SUBSET-SUM} \leq_p \text{SCHEDULE-RELEASE-TIMES} \).

**Pf.** Given an instance of \( \text{SUBSET-SUM} w_1, \ldots, w_n \), and target \( W \),

- Create \( n \) jobs with processing time \( t_i = w_i \), release time \( r_i = 0 \), and no deadline \( d = 1 + \sum_j w_j \).
- Create job 0 with \( t_0 = 1 \), release time \( r_0 = W \), and deadline \( d_0 = W+1 \).

Can schedule jobs 1 to \( n \) anywhere but \([W, W+1]\)

---

8.10 A Partial Taxonomy of Hard Problems
Polynomial-Time Reductions

3-SAT

INDEPENDENT SET

VERTEX COVER

SET COVER

HAM-CYCLE

TSP

DIR-HAM-CYCLE

3D-MATCHING

GRAPH 3-COLOR

SUBSET-SUM

PLANAR 3-COLOR

Dick Karp (1972) 1985 Turing Award

Packaging and covering, sequencing, partitioning, numerical constraint satisfaction

3-SAT reduces to INDEPENDENT SET