

## Problem Set 6 - NP-Complete Reductions

### 1. Reductions

#### (1a)

#### Hamiltonian Path $\leq$ Hamiltonian Circuit

Modify your graph by adding another node that has edges to all the nodes in the original graph.

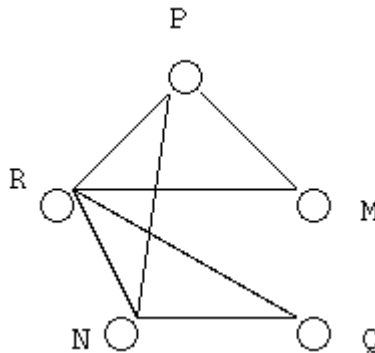
If the original graph has a Hamiltonian Path, the new graph will have a Hamiltonian Circuit: the circuit will run from the new node to the start node of the Path, through all the nodes along the Path, back to the new node.

If the original graph does not have a Hamiltonian Path, there can be no Hamiltonian Circuit in the new graph:

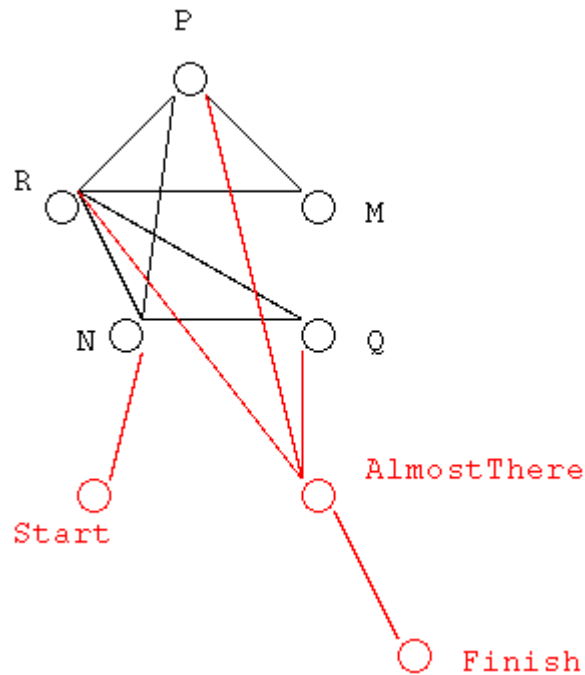
- (1) There is obviously not one starting from the new node. (No edge from the new node can lead to a Path through the graph which allows a return to the new node.)
- (2) There is no possible Circuit starting from any node in the original graph. This is because, at best, the new node would create a Path, but not a Circuit, in the new graph. If there is no Path in the original, there are at least two "gaps" between nodes that would have to be bridged to create a Circuit. Adding the new node could only, at best, bridge one of these, to create a Path but not a Circuit.

#### Hamiltonian Circuit $\leq$ Hamiltonian Path

Modify your graph, say:



and add new nodes thus:



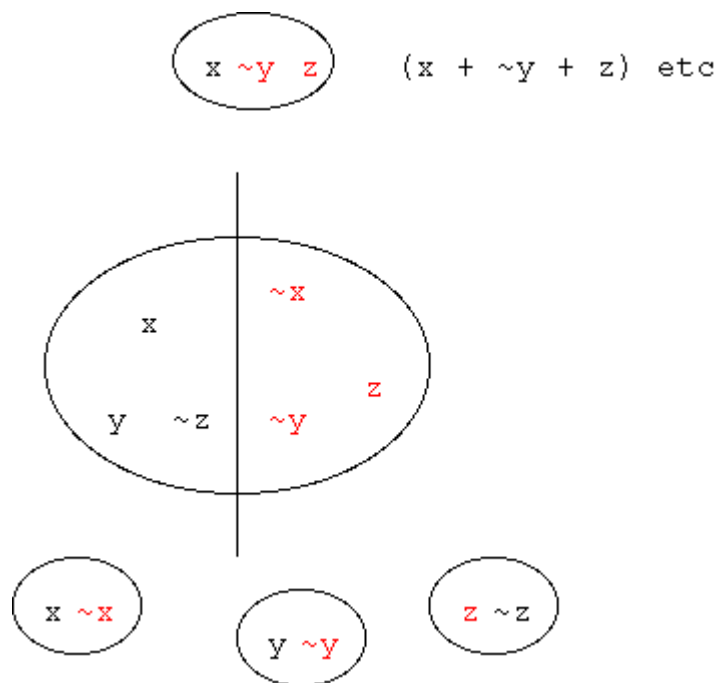
The Start node can be connected to any node in the original graph (in this case N); the AlmostThere node must be connected to all nodes originally connected to N.

If there is a Circuit in the original graph, there will be a Path in the new graph from Start to Finish: follow the Circuit from N to the last node in the Circuit before returning to N (either P, Q or R in the example), then go through AlmostThere to Finish.

If there is no Circuit in the original graph, there can be no Path from N to P, Q or R, so there can be no Path from Start to Finish.

### (1b) Not All Equal 3SAT $\leq$ Set Splitting

Make the set-to-be-split contain all the variables in the NAE3SAT formula and their negations. For each variable-negation pair  $((x, \sim x), (y, \sim y)$  etc) make a subset. Also make a subset for each clause (eg  $(x + \sim y + z)$ ). The set-to-be-split can be split in the required way if and only if the formula has a truth-value assignment that makes it true.



One part of the split set represents "true", the other "false". Each subset must contain one "true" variable and one "false" variable, which is exactly what the satisfiability problem requires.

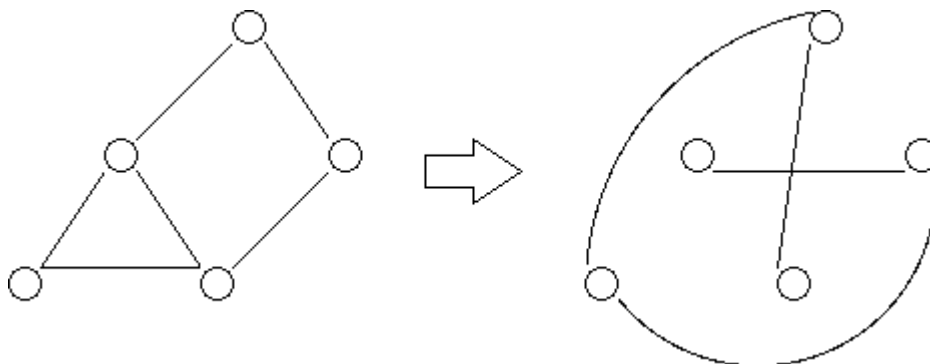
## 2. More Reductions

### (2a) Vertex Cover $\leq$ Independent Set

All of the nodes that are NOT in a graph's minimum vertex cover form that graph's maximum independent set. The independent set can't be smaller, because certainly no two nodes in it are connected (if they were, one of the nodes would have to be in the vertex cover). The independent set can't be bigger, because each node in it is connected to one of the vertex cover nodes.

### (2b) Independent Set $\leq$ Clique

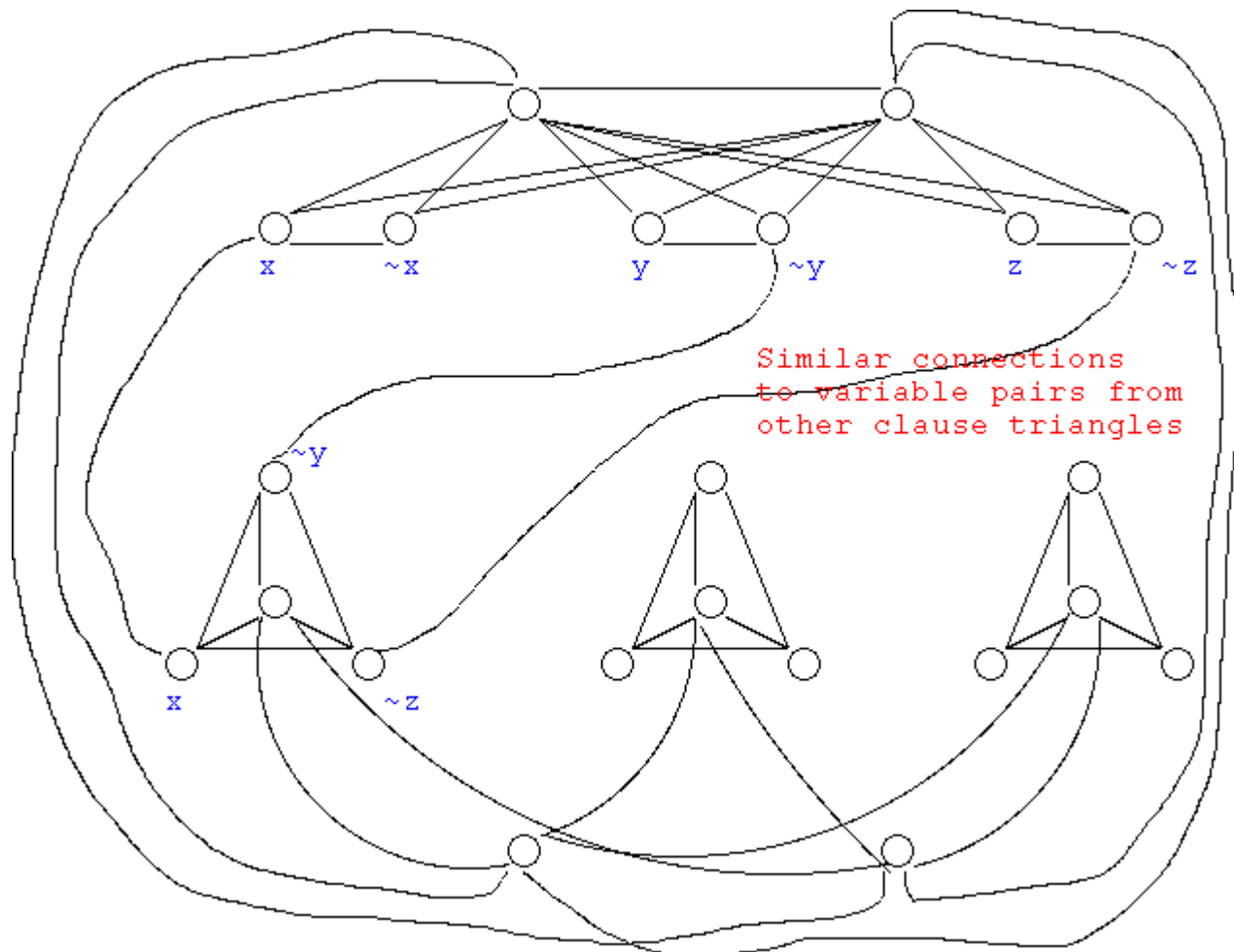
Construct a new graph that "inverts" the original: ie has edges where there weren't any and doesn't have them where there were; eg



A set of nodes is an independent set in one graph if and only if it is a clique in the other, so finding the maximum clique in the new graph finds the maximum independent set in the original.

### (2c) NAE3SAT $\leq$ 4-colorability

This figure is rather complicated. It was invented by several members of the class (not including me - I have merely written it up):



Some points to note about a possible 4-coloring for this:

The two nodes at the top are connected to all of the variable nodes and to each other, so they must each be a different color, and no variable can be either of those colors. Call these two colors the Dummy colors.

Each variable is connected to its negation as well as to the two top nodes, so we must have two other colors, not the Dummy colors, to color the variables. Call these the Truth-value colors. No variable and its negation can have the same Truth-value color.

The two bottom nodes are each connected to the two top nodes, and to each other, so they must use both of the Truth-value colors.

Each of the three outside nodes in the 4-node clause triangles connects to a variable node (so that each variable in the corresponding clause connects to one of these nodes). The middle node of each triangle is connected to each bottom node, so the middle node must be a Dummy color. This leaves the other Dummy color and the two Truth-value colors as candidates for coloring the three outside nodes. (All four colors must be used, as each clause triangle is fully connected.) If the outside nodes are connected to variables that all have the same Truth-value color, we cannot use this color in the triangle, so there is no successful coloring. So in any successful coloring, the outside nodes must be connected to variables of both Truth-value colors.

Bearing these points in mind, convince yourself that the given formula is NAE3-satisfiable if and only if there is a 4-coloring for the corresponding graph.

### 3. The Clique Problem

(3a)

Check every subset of 4 nodes to see if they're all connected to each other. (Dumb but polynomial...)

### (3b)

There are  $n$ -choose-4 subsets, or  $O(n^4)$ . For each subset, there are (at worst)  $4(n-1)$  edges to check, or  $O(n)$ . This is  $O(n^5)$ . (This is using adjacency list representation. With matrix representation, the number of edges to check for each subset is constant, which would give  $O(n^4)$ .)

### (3c)

No planar graph can have a subgraph homeomorphic to  $K_5$  (by Kuratowski's Theorem). So the maximum clique for a planar graph is at most 4. We have a polynomial-time algorithm for finding size-4 cliques (and size 3, and size 2). So clique for planar graphs is not NP-Complete.

## 4. The Coloring Problem

No node in a graph of maximum degree  $\leq 2$  can be connected to more than 2 other nodes, so its minimum coloring can be at most 3 colors. We can, eg, DFS for strongly connected components (certainly polynomial). Each component is either an isolated node, a chain, or a ring. We can also check for cycles in such components in polynomial time. If a component is a ring, count the nodes: an odd number needs 3 colors, an even number only 2. If a component is a chain (more than 1 node, but no cycles) it needs 2 colors. If it's an isolated node (degree 0) it needs 1 color. A particular graph's minimum coloring is the number of colors required by its most color-hungry strongly connected component.

## 5. Partition Revisited

### (5a)

Partition when the sum of the numbers is a perfect square is still NP-Complete. We can reduce ordinary Partition to Perfect Square Partition:

First: of course odd sums don't have a solution.

For even sums, if we have:

$$(1) x_1 + x_2 + \dots + x_i = n,$$

we can convert this to:

$$(2) y_1 + y_2 + \dots + y_i = n^2, \text{ where } y_1 = n \cdot x_1, y_2 = n \cdot x_2 \text{ etc.}$$

If (1) has a partition = to  $n/2$ , (2) has the corresponding partition = to  $n^2/2$ . If (1) does not have such a partition, neither does (2).

### (5b)

#### Partition $\leq$ Subset Sum

Obviously. Set  $B = 1/2 * \text{SUM}(s(a))$ .

#### Subset Sum $\leq$ Partition

Find some number  $Q$  such that  $B + Q = 1/2(n + Q)$ , ie  $Q = n - 2B$ , where  $n = \text{SUM}(s(a))$ .

Add  $Q$  to the original set, the sum of which is now  $n+Q$ . We can partition the new set if and only if there is some subset of numbers in the original set that sums to  $B$ :

(Subset sum of original set) implies (partition of new set): The set that sums to B, with Q, form a partition.

(Partition of new set) implies (subset sum of original set): There are two partitions, one that contains Q and one that doesn't. As  $Q + B =$  the partition sum, all the other numbers in Q's partition sum to B.

The reduction of Partition to Subset Sum implies that Subset Sum is NP-Complete in general because Partition is NP-Complete in general. However, Partition, which is a special case of Knapsack, can be solved in pseudo-polynomial time; therefore, given the reduction of Subset Sum to Partition, so can Subset Sum.

## 6. The ADU Seating Problem

### (6a)

Input: directed graph (nodes are students; arrows from a student to any student that student can stand).

Question: Are there two Hamiltonian Circuits in this graph, such that one exactly reverses the order of the nodes in the other?

### (6b)

Hamiltonian Circuit for any graph with nodes  $\leq$  degree 3 is NP-Complete.

### (6c)

From any node, DFS until the first previously seen node is reached. If you've hit all the nodes by this time, you've got a Hamiltonian Circuit. Check the nodes in reverse order to see if there's also a Circuit the other way.

-----

Erica Klempner  
February 2001