

Reasoning About the Complexity of the Subproblems of a Given NP-Complete Problem

- It is not infrequently the case that even when a general problem is NP-Complete, its various subproblems are amenable to solution by polynomial time algorithms.
- You define subproblems of a problem by placing restrictions on its parameters.
- For example, the subproblems of a graph-based problem may be obtained by placing restrictions on the maximum degree allowed at a vertex, on whether or not the graph is planar, on the connectivity properties of the graph, etc.
- As we create various subproblems by restricting a general problem in different ways, we will end up with some that can be solved in polynomial time, some that are provably NP-complete, and the rest whose time-complexity status cannot easily be ascertained.
- For example, when we consider the three graph-based problems mentioned below and their subproblems obtained by placing restrictions on vertex degree, we obtain the P and NP-Complete subproblems as shown:

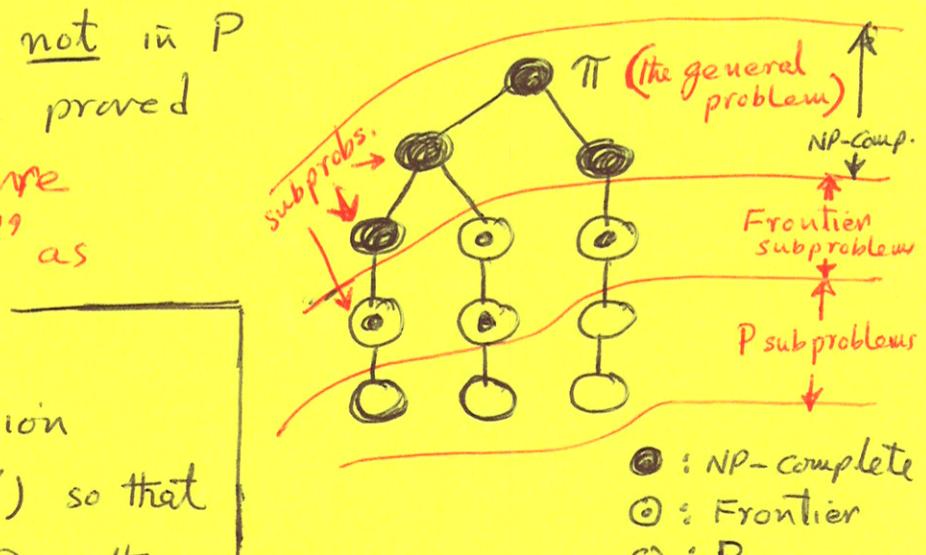
	Subproblems are in P for $D \leq 0$	Subproblems are NP-Complete for $D \geq 1$
VERTEX COVER	2	3
HAMILTONIAN CIRCUIT	2	3
GRAPH 3-COLORABILITY	3	4

$D = \text{Max. degree of a vertex}$

- For each ~~problem~~ problem listed above, when we consider the subproblems corresponding to different max vertex degrees, the subproblems are either in P or are NP-Complete. More generally, though, that is not the case. More generally, when you delineate the subproblems of a problem, you will also end up with subproblems that are not in P and that, at the same time, cannot be proved to be NP-Complete. Such subproblems are referred to as the "frontier subproblems," as shown at right.

- More formally, by the subproblem of a decision problem $\Pi = (D, Y)$ we mean $\Pi' = (D', Y')$ so that $D' \subseteq D$ and $Y' = Y \cap D'$. Recall that D is the set of all instances and Y the set of yes-instances.

- Sometimes the NP-Completeness of a subproblem can be proved by constructing a polynomial transformation from the general problem to the subproblem. The SAT \times 3SAT transformation for proving the NP-Completeness of 3SAT is an



@ : NP-Complete
@ : Frontier
O : P

example of this. 3SAT is obviously a subproblem of SAT.



- In the rest of this lecture we will focus on the general problem of GRAPH 3-COLORABILITY and its subproblems.

GRAPH 3-COLORABILITY (3COL)

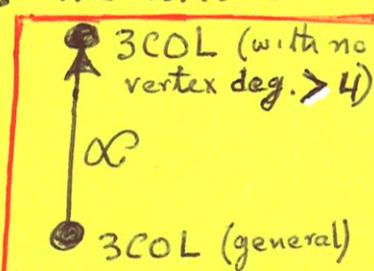
INSTANCE : Graph $G = (V, E)$

QUESTION : Is G 3-colorable? That is, does there exist a function $f: V \rightarrow \{\text{Red, Green, Blue}\}$ such that $f(u) \neq f(v)$ whenever $\{u, v\} \in E$?

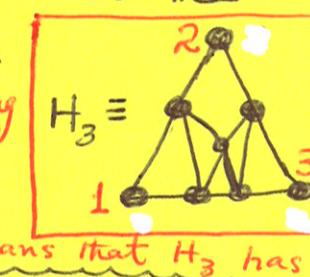
- My goal here is only to discuss the subproblems of 3COL. This problem, which is of some interest to map-makers, was proved to be NP-Complete in 1973 by Stockmeyer. (In map making, you want the adjacent regions — states, countries, etc. — to be colored differently.) 3COL is one of the more important NP-complete problems because it ~~s~~ frequently serves as a base problem for proving the NP-completeness of other problems. Most recently, I saw 3COL being used in that manner in a paper on wireless sensor networks.

THEOREM : 3COL with no vertex degree exceeding 4 is NP-Complete

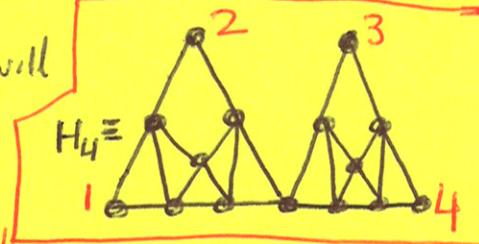
PROOF : We will use the technique of Proof by Local Replacement. We will show how an arbitrary instance of 3COL can be transformed into an instance of the same but with the property that no vertex degree exceeds 4.



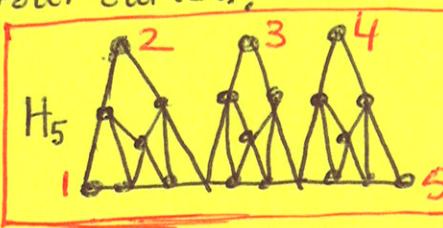
- Let $G = (V, E)$ represent an arbitrary instance of 3COL and $G' = (V', E')$ the corresponding instance after the polynomial transformation. The following graph, which we will denote by H_3 , plays an important role in this transformation. H_3 is 3-colorable, but with every way of 3-coloring H_3 assigning the same color to the three corner vertices labeled 1, 2, and 3. These will be called the outlet vertices. The suffix 3 in H_3 means that H_3 has 3 outlet vertices.



- Now let's construct H_4 by joining two H_3 graphs. This we will accomplish by joining the last outlet of the first H_3 with the first outlet of the second H_3 . H_4 retains the curious property of H_3 that every way of 3-coloring H_4 assigns the same color to the four outlet vertices shown. This can be verified by inspection, although it is also a logical consequence of the fact that whatever color is given to the vertex that is common to the two H_3 structures will be the color of all four outlets.



- We can now construct H_5 in the same manner, that is by extending H_4 with another H_3 . This leads to a recursive definition for H_k for any $k \geq 4$. H_k is obtained from H_{k-1} by joining the last outlet of H_{k-1} with the first outlet of a fresh copy of H_3 .



- We transform G into G' as follows: We replace every vertex v of degree $k > 4$ in G by H_k . We join the k outlets of H_k to the rest of the vertices of G in the same manner as v . For example, if G :



$$\text{then } G' =$$



- It is obvious from the construction that G is 3-colorable iff the mapped G' is 3-colorable.

THEOREM : Planar graph 3-colorability is NP-Complete.

PROOF : This proof is also based on local replacement. The crossover points of a nonplanar graph are replaced by the H structure. For the 4 outlets shown, x, x', y, y' , any 3-colorability assignment f satisfies $f(x) = f(x')$ and $f(y) = f(y')$. See G+J for the rest of the proof.

