day 26 4/20/98

1 Complexity theory!

1.1 definitions

- CNF: an and of or's. Example: $(x_1 \bigvee x_2 \bigvee x_3) \wedge (x_1 \bigvee x_4) \wedge (x_3 \bigvee x_2)$
- clique: a group of fully connected nodes in a graph

2 np completeness

- The 'can I satisify this CNF' problem is NP complete.
- 'Is there a clique of size k' is also NP complete.

3 reduction

The CNF problem can be transformed into a clique problem by a polynomial transformation. This means, $CNFSat \leq_m^p Clique$ or 'CNFSat reduces to clique'.

A common goal in complexity theaory is reducing one problem to another in polynomial time. Basically, by reducing known NP-complete problem X to problem Y by a polynomial transformation. Then Y is 'at least as hard as X' which mean Y is NP-hard.

The Clique problem can als be reduced to the CNF problem, implying $Clique \leq_m^p CNF$. So

$Clique \equiv_m^p CNF$

4 definitions

- $A \leq_m^p B$ means A can be solved by calling B once plus some polynomial work.
- $A \leq_m^p B$ and $B \leq_m^p C$ then $A \leq_m^p C$
- $A \leq_T^p B$ The same as before except B can be called polynomial times.
- $P =$ set of problems solvable in polynomial time
- $NP =$ set of problems checkable in polynomial time for 'yes' answers.
- $NP - Complete =$ Can't? solve the problem in polynomial time.
- $NP - Hard = NP - Complete$ or harder.

If a problem is $NP - Complete$ then every NP problem can reduce to it.

The big question in computer science is 'does $NP = P$'?

- $Co - NP$ = you can check 'No' answers in polynomial time.

factoring is in $Co - NP$ and NP.

5 reductions

5.1 $3CNFSat \leftrightarrow CNFSat$

- $3CNFSat$ problems have the additional constraint that the CNF is an and of 'ors' of at most 3 terms.

We have $CNFSat \geq_m^p 3CNFSat$ automatically. So, let's show $CNFSat \leq_m^p 3CNFSat$

- general fact: $(A \bigvee x) \wedge (\overline{x} \bigvee B) \Leftrightarrow (A \bigvee B)$ (resolution rule)

Using the resolution rule, $CNFSat$ can be reduced to $3CNFSat$ by repeated application.

5.1.1 $2CNF$

2CNF is easy to solve. $x_1 \bigvee x_2$ means $\overline{x_1} \Rightarrow x_2$ so the terms of $2CNF$ can be turned into a graph of implications on which it is easy to solve the $2CNF$ problem. In fact, it can be solved in linear time.

5.2 independent set $\leftrightarrow$ clique

- independent set = are there k nodes with no edges between any of them on a graph?

The reduction is done by solving the complement of the graph.

5.3 Vertex cover $\leftrightarrow$ independent set

- Are there k nodes which touch every edge?

All of the nodes not in the vertex cover are in the independent set.

Vertex Cover $\equiv_m^p$ Independent Set

5.4 CNFSat $\equiv_m^p$ 3 colorability

- Can this graph be colored with 3 colors s.t. no edge has the same color at each end?

5.4.1 CNFSat $\leq_m^p$ 3 colorability

$(x_1 \bigvee \overline{x_2} \bigvee x_3 \bigvee \overline{x_4} \bigvee x_5) \wedge ...$

Draw a graph where every variable is forced to be 'red' or 'green'. Then build a graph with several constraints.