NP-complete Problems

CS255

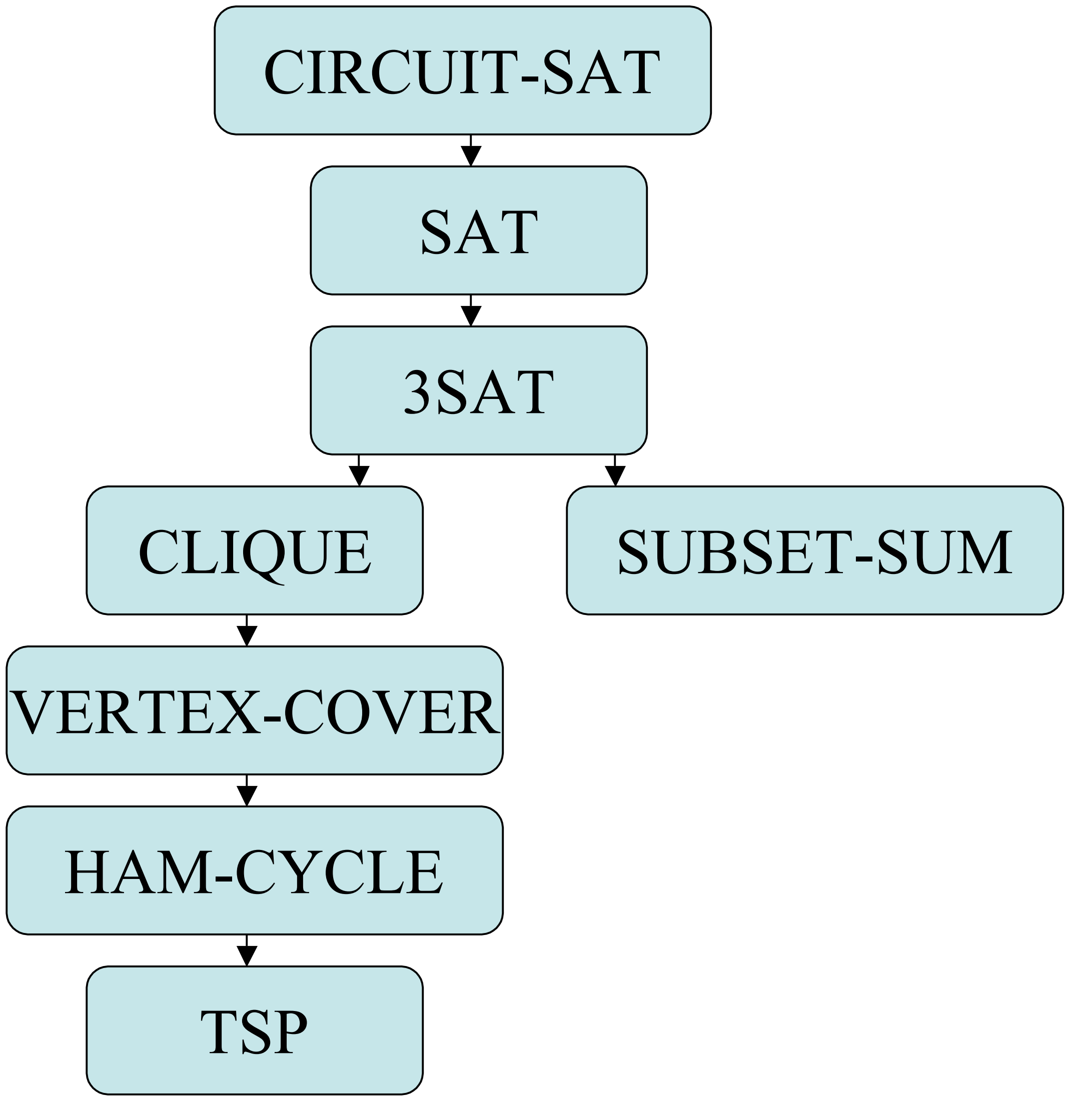
Chris Pollett

May 1, 2006.

Outline

- Some NP-Complete Problems

Reductions we will show



SAT and 3SAT

- These languages were both defined last day.

Theorem. Both *SAT* and *3SAT* are **NP**-complete.

Proof. First both languages are in **NP** by the same argument that showed *CIRCUIT-SAT* in **NP**. Given an instance <*C*> of *CIRCUIT-SAT*, let gate *i* be coded as <*i*, *type*, *j*, *k*>. Here *type* is AND, OR, NOT, or input, and j, $k < i$ are gates which are inputs to this gate. A 0 for j or k means that argument is not used. Let c_i*'s* be new variables other than the input variables x_j. Recall the symbol ⇔ is true if both its boolean inputs have the same value. For each gate we create a boolean formula either of the form $c_i \Leftrightarrow (c_j\ type\ c_k)$, where *type* is replaced with AND or OR; or of the form $c_i \Leftrightarrow (type\ c_j)$ in the case of NOT or an input (in the latter case *type* is nothing). The *SAT* formula we output on input <C> is the conjunction of all such defining formulas conjuncted with c_w, where *w* is the last gate in the formula. The idea is if c_w is true, then its defining equation $c_w \Leftrightarrow \ldots$ must be true and this propagates back to some setting of the leaves which will make the circuit true. By rewriting each $c_i \Leftrightarrow (c_j\ type\ c_k)$ formulas in 3CNF we can make this whole formula into 3CNF. We can pad clauses with less than 3 literals with dummy variables to make all clauses the same size.

CLIQUE

- A **clique** in an undirected graph $G=(V, E)$ is a subset $V' \subseteq V$ of vertices each pair of which is connected by an edge.
- $CLIQUE = \{<G, k> \mid G$ is a graph with a clique of size $k\}$

Theorem. *CLIQUE* is **NP**-complete.

Proof. First *CLIQUE* is in **NP** because we can just guess a set of more than k vertices and check for each possible edge.Next we show how to reduce *3SAT* to *CLIQUE*. Let $F = C_1 \wedge C_2 \ldots \wedge C_k$ be an instance of *3SAT*… For each $C_r = (l^r_1 \vee l^r_2 \vee l^r_3)$, we put a triple of vertices into our graph G, v^r_1, v^r_2, v^r_3. We put an edge between vertices v^r_i and v^s_j if they are in different triples and their corresponding literals are not negations of each other. Let (G, k) be the output instance of *CLIQUE*. Notice if there is a satisfying assignment to F then if we look at the corresponding vertices v^r_j of at least one satisfied literal/clause, it will be a *CLIQUE* of size k in G. As there are no edges between vertices coming from the same clause. Any *CLIQUE* of size k has to have at least one vertex v^r_j for each $1 \leq r \leq k$. Choosing an assignment so that the corresponding literals for each v^r_j evaluates to true gives a satisfying assignment. So the reduction works.

VERTEX-COVER

- A **vertex cover** in an undirected graph *G*=(*V*, *E*) is a subset of the vertices *V′* such that each vertex in *V* is connected to a vertex in *V′* by an edge.
- VERTEX-COVER={<*G*, *k*>: graph *G* has a vertex cover of size *k*}

Theorem. VERTEX-COVER is **NP**-complete.

Proof. To see it is in **NP** notice if we guess a set of *k* edges we can check if it is a vertex in polynomial time. To see it is **NP**-complete we reduce *CLIQUE* to this problem. Let $\overline{G}$ denote the complement of a graph *G*=(*V*, *E*), that is, the graph with the same vertices, but with edges {*i*, *j*} iff {*i*, *j*} is not an edge of *G*. Then notice *G* has a clique of size *k* iff $\overline{G}$ has a vertex cover of size |*V*|-*k*.