ECM3404: Logic and Computation

Tutorial, 5th May 2009

NP-completeness and Problem Reduction

Cook's Theorem shows how any NP problem can be reduced to SAT in polynomial time. In the lectures we saw how this can be done in general by using clauses to describe the Turing machine which does the certificate-checking for the problem. This assures us that polynomial-time conversion is always possible, but it does not necessarily give us the most natural or efficient conversion in every case.

In this exercise you are asked to devise a way of reducing the **Hamiltonian Circuit Problem** (HAM) into SAT directly, without making use of the Turing machine associated with the problem.

HAM may be stated as follows:

- *Input*: a graph (G, V) where
  - $V = \{v_1, v_2, \ldots, v_n\}$ is the vertex-set, and
  - $E = \{e_{ij} \mid v_i \text{ is joined to } v_j\}$ is the edge-set.
- *Output*: 'yes' iff there exists a permutation $v_{i_1} v_{i_2} \ldots v_{i_n}$ of the vertices such that for $j = 1, \ldots, n-1$, $e_{i_j i_{j+1}} \in E$, and also $e_{i_n i_1} \in E$.

The size of an input instance is the number of vertices plus the number of edges ($|V|+|E|$).

We need to encode any given instance I of HAM as a set of propositional clauses that is satisfiable iff I is a positive instance. Moreover, we want the total number of literals in the clause-set to be bounded by a polynomial function in the size of the instance.

1. Topic for discussion: how can we do this?
2. When you've decided how to do it, actually write down the clauses needed for the following simple instance of HAM, and show that they are unsatisfiable: