CSE 347 Analysis of Algorithms February 25, 2019

Lecture 11 Summary

Instructor: Brendan Juba

1. Reductions allow us to relate the existence of polynomial-time algorithms for problems *regardless* of whether or not we actually know such algorithms
2. The independent set problem has equivalent optimization and decision versions
   (a) An *independent set* in a graph $G = (V, E)$ is a set $S \subseteq V$ such that no pair of vertices in S is joined by an edge.
   (b) The natural optimization problem is: given G, find an independent set of maximum size.
   (c) A *decision* version of the problem is: given G and an integer k, does G have an independent set of size at least k?
   (d) The optimization and decision versions of the problem can be reduced to each other, so they are equivalent.
      i. Given the solution to the optimization problem, we check if it has size at least k
      ii. If we can solve the decision version, we can find the largest size k and then rule out vertices one at a time (by adding edges) until we find a set S that contains precisely vertices that must be in any independent set of size k on the remaining vertices. This will be an independent set of size k.
3. The vertex cover problem is equivalent to the independent set problem.
   (a) A *vertex cover* in G is a set $S \subseteq V$ such that every edge has at least one endpoint in G.
   (b) Vertex cover similarly has equivalent optimization and decision versions
   (c) Lemma: S is an independent set iff $V \setminus S$ is a vertex cover
   (d) Therefore, there is an independent set of size k iff there is a vertex cover of size $|V| - k$. This gives an easy reduction in both directions.
4. 3-SAT reduces to independent set, even though they look very different
   (a) A *conjunctive normal form* formula (CNF) is an AND of *clauses*. A clause is an OR of *literals*. A literal is a Boolean variable or the negation of a Boolean variable. If the clauses of the CNF all have at most k literals, it is a k-CNF.
   (b) The *satisfiability problem* is, given a CNF φ, is there a setting of the Boolean variables that will cause φ to evaluate to true? (A *"satisfying assignment"* to φ.) k-SAT is the satisfiability problem for k-CNFs only.
   (c) We can reduce 3-SAT to independent set via a "gadget reduction"
      i. A satisfying assignment to a CNF identifies at least one literal in each clause that is made true by the assignment

ii. It's enough to just identify which literals are satisfied in each clause: that is, we find a set of literals, one per clause, such that we don't pick both a Boolean variable and its negation in different clauses.

iii. Edges in an independent set problem enforce such mutual exclusion: we create a variable for each literal in the formula and join variables in the same clause with edges (enforcing one literal per clause) and join pairs of negated variables (enforcing consistency)

iv. We can read off a satisfying assignment from an independent set in this graph, and conversely, from a satisfying assignment can identify an independent set. Therefore, to solve satisfiability, it's enough to solve the independent set problem on this graph.