TU Dresden, Fakultät Informatik
Daniel Borchmann

Summer Term 2016

Advanced Topics in Complexity Theory

Exercise 6: Optimization Problems

2016-05-10

(This exercise is based on Section 17.1 of Papadimitriou's book *Computational Complexity*.)

Assessing the exact complexity of optimization problems is often rather difficult. For example, up to now we were unable to give an exact classification of the Traveling Salesperson Problem (TSP), in the sense of providing a natural complexity class for which TSP is complete. In this exercise, we want to investigate some aspects of this problem.

We shall start with a modification of TSP, called ExactTSP: given a complete weighted graph G and some $B \in \mathbb{N}$, is the length of the shortest tour in G exactly B?

Exercise 6.1 Argue that ExactTSP can be represented as an intersection of $\mathsf{TSP}_\mathsf{D} \in \mathsf{NP}$ and $\mathsf{TSPcompl}_\mathsf{D} \in \mathsf{coNP}$, asking whether the shortest tour in a graph has length *at least* some given threshold B.

This motivates the following definition.

Definition 6.2 The class DP consists of all languages L such that there exist two languages $L_1 \in \mathsf{NP}$ and $L_2 \in \mathsf{coNP}$ such that $L = L_1 \cap L_2$. ♢

The name DP derives from "difference", as $L \in \mathsf{DP}$ if and only if there exist two languages $L_1, L_2 \in \mathsf{NP}$ such that $L = L_1 \setminus L_2$.

Note that DP is *not necessarily* the same class as $\mathsf{NP} \cap \mathsf{coNP}$, i.e., they are not known to be equal. For example, for the later class there are no known complete problems, while DP easily has some.

Exercise 6.3 Denote with SATUNSAT the following problem: given two Boolean expressions φ and φ' in conjunctive normal form, is it true that φ is satisfiable and φ' is not?

Show that SATUNSAT is DP-complete.

We next want to argue that ExactTSP is DP-complete. The main part for this is to show how to reduce SATUNSAT to ExactSAT. For this we use the following fact.

Lemma 6.4 There exists a polynomial-time reduction f from 3SAT to HamiltonianPath such that whenever φ is a 3CNF formula, then $f(\varphi)$ contains a *broken Hamiltonian path*, i.e., two node-disjoint paths that cover all nodes in the target graph. Those two paths are connected via an edge if and only if φ is satisfiable.

Given two Boolean expressions φ, φ', we apply the reduction f to obtain two graphs $G = f(\varphi)$ and $G' = f(\varphi')$. Denote with v_1, v_2 the start and end node in G of the (possibly broken) Hamiltonian path in G, and likewise v_1', v_2' in G'. We obtain a new graph H by identifying v_1 and v_2' as well as v_2 and v_1'.

We next attach weights to H. If $\{ i, j \}$ is an edge in either G or G', it gets assigned the weight one. Otherwise, if both i, j are nodes in G, they get assigned weight 2. All other pairs $\{ i, j \}$ get weight 4.

Exercise 6.5 Show that the shortest tour in H has length exactly $n+3$, where n is the number of nodes in H, if and only if $(\varphi, \varphi') \in \mathsf{SATUNSAT}$. Conclude that $\mathsf{ExactSAT}$ is DP-complete.

To asses the complexity of the actual TSP problem, the class DP is not enough (for all we know). We therefore introduce another class that makes use of *oracle machines.*

Recall that an oracle Turing machine (OTM) is a usual Turing machine M having an extra tape (called the *oracle tape* of M) and three distinguished states $q_?, q_{\text{yes}}, q_{\text{no}}$. If $\mathcal{O} \subseteq \Gamma^*$, then the machine $M^{\mathcal{O}}$ can in addition to the usual operations of a Turing machine *query the oracle* $\mathcal{O}$: whenever $M^{\mathcal{O}}$ reaches state $q_?$, the next state of $M^{\mathcal{O}}$ is q_{yes} if the content of the oracle tape is contained in $\mathcal{O}$ and q_{no} otherwise.

Definition 6.6 The class $\mathsf{FP}^{\mathsf{NP}}$ consists of all functions $f\colon \Sigma^* \to \Sigma^*$ that are computable in polynomial time by an OTM with an oracle for SAT. $\diamondsuit$

It turns out that TSP is complete for $\mathsf{FP}^{\mathsf{NP}}$. Showing this, however, is beyond of the scope of this exercise. Instead, we shall show that another problem is $\mathsf{FP}^{\mathsf{NP}}$-complete.

Let us define the problem $\mathsf{MaxOutput}$ as follows: Let N be a non-deterministic Turing machine that on inputs 1^n halts after $\mathcal{O}(n)$ steps with a binary string of length n on its output tape. What is the largest output, considered as a binary number, of N on input 1^n?

Exercise 6.7 Show that $\mathsf{MaxOutput}$ is in $\mathsf{FP}^{\mathsf{NP}}$.

Showing completeness of $\mathsf{MaxOutput}$ for $\mathsf{FP}^{\mathsf{NP}}$ is a bit more involved. Let $F \in \mathsf{FP}^{\mathsf{NP}}$, i.e., there exists a polynomial-time deterministic OTM M such that for all $x \in \Sigma^*$ we have $M^{\mathsf{SAT}}(x) = F(x)$. Here, $M^{\mathsf{SAT}}(x)$ denotes the output of M^{SAT} on input x.

We need to give a reduction from F to $\mathsf{MaxOutput}$. For this we need to provide two logspace-computable functions r, s such that

- for each $x \in \Sigma^*$, $r(x)$ is an instance of $\mathsf{MaxOutput}$, and
- if y is the maximal output of $r(x)$, then $s(y) = F(x)$.

The reduction r works as follows: let p be the bounding polynomial of M^{SAT}. On input x, define $n = p(|x|)^2$. Then N on input 1^n first write x on another tape and simulates M on input x.

Since M is deterministic, this simulation can be carried out except for the oracle queries of M. If N encounters such a query, it guesses the answer z_1 to that query: either $z_1 = 1$ if the query is satisfiable, and $z_1 = 0$ otherwise. If $z_1 = 0$, then N continues the simulation of M. However, if $z_1 = 1$, then N guesses a satisfying assignment of the query and checks it. If the check succeeds, N continues the simulation. Otherwise, it outputs 0^n and halts. If the latter happens, we call this an *unsuccessful computation.*

Suppose that N succeeds in simulating $M^{\mathsf{SAT}}(x)$ until the machine halts. Denote with $z_1, z_2, \dots$ the non-deterministic choices of N during this simulation. The output of N is then $z_1 z_2 \dots$ followed by as many zeros are necessary to reach length n, followed by $M^{\mathsf{SAT}}(x)$. This is called a successful computation. Note that a *successful computation* of N may still represent an *erroneous* simulation of M^{SAT} on input x, as N may have guessed the answers to the oracle queries incorrectly.

Exercise 6.8 Show that the successful computation corresponding to the largest output of N on input 1^n corresponds to a correct simulation of M^{SAT} on input x. Conclude that $\mathsf{MaxOutput}$ is complete for $\mathsf{FP}^{\mathsf{NP}}$. (Don't forget to provide the function s!)