Computational Complexity — tutorial 5

Completeness cntd.

NP-completeness. Some well-known NP-complete problems:

- NP Halting Problem. Given a non-deterministic Turing Machine $\mathcal{M}$, input w and an integer k (written in unary), is there an accepting run of $\mathcal{M}(w)$ which terminates in k steps?
- Boolean Satisfiability (SAT). Given a boolean formula φ over boolean variables $x_1, x_2, \ldots, x_n$, is it satisfiable for any values of $x_1, \ldots, x_n$? (That is, is $\exists_{x_1}\exists_{x_2}\ldots\exists_{x_n}\varphi(x_1, \ldots, x_n)$ true?)
- 3-CNF-SAT. Similar to above, but φ is of the form $\varphi = \psi_1 \wedge \psi_2 \wedge \cdots \wedge \psi_m$, and each ψ_i is a disjunction (OR) of 3 literals (x_i or $\neg x_i$). E.g., $\varphi = (x_1 \vee x_3 \vee \neg x_2) \wedge (x_1 \vee \neg x_3 \vee \neg x_4)$.
- Clique. Given an undirected graph G and an integer k, does G contain a clique (a complete graph) on k vertices?
- Independent Set. Given an undirected graph G and an integer k, does G contain an independent set of size k (that is, k vertices without any edges between them)?

PSPACE-completeness. Some well known PSpace-complete problems:

- PSpace Halting Problem. Given a non-deterministic Turing Machine $\mathcal{M}$, input w and an integer k (written in unary), is there an accepting run of $\mathcal{M}(w)$ whose memory usage is bounded from above by k?
- Quantified Boolean Formula. Given a boolean formula φ (with quantifiers) without any free variables (e.g. $\exists x_1 \forall x_2 \exists x_3 (x_1 \wedge (\neg x_2 \vee x_3)))$, is it true? Note that we can assume that all quantifiers occur at the beginning of the formula.

For your enjoyment after the tutorial: G. Aloupis, E. D. Demaine, A. Guo, G. Viglietta. *„Classic Nintendo Games are (Computationally) Hard."* Available at `https://arxiv.org/abs/1203.1895`.

1. Prove that the following computational problems are NP-complete. For simplicity, assume that graphs are given as $n \times n$ adjacency matrices, and all integers on input are given in binary.

(d) Vertex Cover problem: given an undirected graph G and an integer k, is there a subset A of k vertices of G such that each edge is incident to some vertex of A?

(e) **[solved]** $\{0,1\}$-Linear Programming problem (in a decision form): given an integer matrix $A \in \mathbb{Z}^{m \times n}$ and a vector $b \in \mathbb{Z}^m$, is there an **integer** vector $x \in \{0,1\}^n$ such that $Ax \leqslant b$?

($Ax \leqslant b$ is a set of linear inequalities of the form $a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leqslant b_i$.)

If x can be arbitrary $\mathbb{Z}^n$, then the problem is NP-complete, but this is non-trivial. Why?

Note: If x can be arbitrary $\mathbb{R}^n$, then the problem is in P, but the algorithms are hard.

(f) Subset Sum problem: given a multiset of n non-negative integers $\{a_1, \ldots, a_n\}$, and an integer A (target), is there a subset of the multiset whose sum is A?

What if A was written in unary instead of binary?

2. [**implicitly solved last week**] Formula Game is a two-player game where a formula φ with n free variables $x_1, x_2, \ldots, x_n$ is given.

In the first step, the first player chooses the value of x_1. In the second step, the second player chooses the value of x_2. Then, in the third step, the first player chooses the value of x_3 etc. The game ends after n steps. The first player wins if the formula evaluates to `true` after n steps.

Prove that determining which player wins Formula Game on a given formula is PSpace-complete.

3. Generalized Geography is a two-player impartial game played on a directed graph G. A token is placed in some vertex $v_0 \in V(G)$. The players move alternately, taking a token from some $v \in V(G)$ and moving it along an edge to another vertex $w \in V(G)$ (so $vw \in E(G)$ must hold). However, the token **cannot** appear multiple times at any vertex. Whoever cannot make a legal move, loses.

Prove that determining which player wins Generalized Geography, assuming both players play optimally, is PSpace-complete.

4. (homework 2017) Let *pattern* be a word consisting of letters and question marks. A pattern p matches a word w if p can be obtained by taking an infix of w and replacing some of its characters by question marks. For instance, `ab?d?` matches `abcde`, `abababdddd`, but not `abcd` or `abaaaadx`. Consider the following problem: given an alphabet Σ and two sets P and N of patterns, decide whether there exists a word $w \in \Sigma^*$ matching all patterns in P and no pattern in N. Prove that this problem is PSpace-complete.

5. In NFA Universality, you're given a non-deterministic finite automaton A over an alphabet Σ, and you need to check if A can produce all words over Σ. Prove that NFA Universality is PSpace-complete.

6. st-Reachability is a decision problem where you're given a **directed** graph G (over vertices $1, \ldots, n$) and two of its vertices $s, t \in [1, n]$, and you're asked if there exists a directed path from s to t in G. Prove that st-Reachability is NLogSpace-complete (with respect to logarithmic space reductions).

Hint: prove it from the definition. We must take an arbitrary problem X in NLogSpace and reduce it to st-Reachability *in logarithmic space. What does it mean that X is in NLogSpace?*

7. In Circuit Value Problem, you're given a binary circuit with an n-bit input and one output, along with the input $x_1, x_2, \ldots, x_n$, and you're asked if the output of the circuit for that input is `true`. Prove that Circuit Value Problem is P-complete (with respect to logarithmic space reductions).