Practical sessions for Introduction to Complexity and Computability

January 7th, 2020

Extra homeworks

Each of the following problems is for 1 point if not stated otherwise. You can choose only some subproblems to obtain a corresponding proportion of a point.

1. Construct a Turing machine M (with all details) that
   (a) transforms natural numbers in binary encoding (over the alphabet $\{0,1\}$) into quaternary encoding (over the alphabet $\{0,1,2,3\}$),
   (b) accepts precisely the words of zero's whose length is a power of 2, i.e. $L(M) = \{0^{(2^n)}\}$.
2. Show (with all details) that for every TM
   (a) there is an equivalent TM (up to encoding of the input) with only a single symbol alphabet (aside from λ),
   (b) there is an equivalent pushdown automata with two stacks (and without input tape).
3. (0.5 point) Show that the following language is partially decidable
$$SIZE = \{\langle M, k\rangle \mid |L(M)| \geq k\}.$$
4. (0.5 point) Let A be a partially decidable language. Show that A is decidable if $A \leq_m \overline{A}$.
5. Let us define disjoint union $\oplus$ of two languages A and B over the alphabet $\{0,1\}$ by
$$A \oplus B = \{a0 \mid a \in A\} \cup \{b1 \mid b \in B\}.$$
   (a) Show that $A \leq_m A \oplus B$ and $B \leq_m A \oplus B$.
   (b) Show that if C is a language over $\{0,1\}$ such that $A \leq_m C$ and $B \leq_m C$, then also $A \oplus B \leq_m C$.
6. Find a partial computable function $g\colon \mathbb{N} \to \mathbb{N}$ such that dom $\varphi_{g(x)} = \{x\}$ for every $x \in \mathbb{N}$.
7. Show that the following languages are undecidable.
   (a) $A = \{\langle M, N\rangle \mid L(M) \setminus L(N) \neq \emptyset\}$
   (b) $B = \{\langle M, x\rangle \mid M(x)\downarrow$ and the input tape is empty after M halts$\}$
8. Show that the following problems are undecidable.
   (a) Does the given program return 0 for some input?
   (b) Do the given two programs compute the same arithmetic function?
9. For each pair of complexity classes below decide if there is any inclusion between them, and if it is a proper inclusion (i.e. $\subsetneq$) (by using theorems and propositions from the lecture). Note that some relations may not be known.
   (a) SPACE(n^3) and TIME(2^{n^3})
   (b) TIME(2^n) and NSPACE($\sqrt{n}$)
   (c) NSPACE($(\log n)^3$) and SPACE(n)
   (d) NTIME(n^3) and SPACE(n^6)

10. A propositional formula in DNF is a disjunction of (any number of) elementary conjunctions, an *elementary conjunction* (also called a *term*) is a conjunction of (any number of) literals, a *literal* is a propositional variable or its negation.

    Decide (and show why) which of the following problems are in **P**, which are in **NP**, resp. in **coNP**. Note that some answers may not be known. Given a formula φ in DNF,

    (a) Is φ satisfiable?

    (b) Is $\neg\varphi$ satisfiable?

    (c) Is φ valid (i.e. a tautology)?

    (d) Is φ contradictory (i.e. $\neg\varphi$ is valid)?

11. (0.5 point) Decide (and show why) whether the following problem is in **P** or is **NP**-complete.

    Instance: A formula φ in CNF.

    Question: Are there two different satisfying assignments of φ?

12. (0.5 point) Show that if there exists a polynomial algorithm deciding SAT problem, then there is also a polynomial algorithm for finding a satisfying assignment (if it exists) for CNF formulas.

13. Show that the following problems IP (integer programming) and BIP (binary integer programming) are **NP**-hard by polynomial reductions of some of the above mentioned problems.

    Instance (the same for IP and BIP): An $m \times n$ matrix A over integers and a vector b of m integers.

    Question (IP): Is there a vector x of n integers such that $Ax \geq b$?

    Question (BIP): Is there a vector $x \in \{0,1\}^n$ such that $Ax \geq b$?

14. In the optimization version of the vertex cover problem (Min-VC) we ask for a vertex cover of minimal size for a given graph G. Consider the following (greedy) approximation algorithm.

    For an arbitrary edge uv of G we add both u, v to a set S (initially empty) and we remove all edges incident to u or v from G. We repeat this step until G is empty. Then we output S as a vertex cover of G.

    Determine the approximation ratio of this algorithm.

15. In the Max-SAT problem we ask for an assignment of a given CNF formula that satisfies maximal number of clauses. Show that there is a 2-approximation algorithm running in polynomial time for this problem.