Computational Complexity: Take Home exam. Submission date 15/7/2017.

Teacher: Ronen Shaltiel

June 19, 2017

Instructions

General:

- You must hand in printed solutions. (I'm willing to accept calculations in handwriting if you don't know how to print them but the rest must be printed).
- Please write clearly and precisely!
- I will not accept solutions that are longer than 12 pages.

Rules:

- This is a test! You are not allowed to collaborate with other people and must do the work on your own. For some of the questions it may be the case that there are solutions on-line. Don't use them!
- Please include in your submission a **signed statement** saying that: I (*include name and ID*) hereby declare that I did not discuss this project with any other people and the solution that I am submitting is my own work. If you submit electronically (which is preferred) make sure to place such a statement in my mailbox. Note that I may want to set up a date in which I will spend 10-20 minutes with every student and have him explain his solutions to me.

Grading:

- You should solve 4 questions.
- Each question has a score. Note that by choosing different subsets of questions you can aim for different final scores.
- Write clear and full answers! A 10 point bonus will be given to submissions which are clear and well written.
- You are allowed to answer one additional question over the 4 questions required (that is you can answer 5 questions), and I will base the scoring on the best 4 questions.

Good Luck!

Definitions from class

Definition 1 (The class MA_s^c)**.** *A language L is in MA_s^c if there exists a polynomial time machine V and a constant c such that for every string x of length n:*

- *If $x \in L$ then $\exists a \in \{0,1\}^{n^c} : \Pr_{r \in_R \{0,1\}^{n^c}}[V(x,r,a) = 1] \geq c$.*
- *If $x \notin L$ then $\forall a \in \{0,1\}^{n^c} : \Pr_{r \in_R \{0,1\}^{n^c}}[V(x,r,a) = 1] \leq s$.*

We define $MA = MA_{1/3}^{2/3}$.

Definition 2 (The class AM_s^c)**.** *A language L is in AM_s^c if there exists a polynomial time machine V and a constant c such that for every string x of length n:*

- *If $x \in L$ then $\Pr_{r \in_R \{0,1\}^{n^c}}[\exists a \in \{0,1\}^{n^c} : V(x,r,a) = 1] \geq c$.*
- *If $x \notin L$ then $\Pr_{r \in_R \{0,1\}^{n^c}}[\exists a \in \{0,1\}^{n^c} : V(x,r,a) = 1] \leq s$.*

We define $AM = AM_{1/3}^{2/3}$.

Questions:

1. (20) (Circuit minimization). Given a function $f : \{0,1\}^\ell \to \{0,1\}$ we can encode it as a string x_f of length $n = 2^\ell$ by considering its truth table. (That is $x_i = f(i)$). Consider the following language:

   $Min{-}Circuit = \{x, s :$ The smallest boolean circuit computing the function f encoded by x is of size $s\}$

   Show that $Min - Circuit \in P^{NP}$.

2. (20) (st-connectivity in $O(1)$-regular graphs of logarithmic diameter) Design an algorithm that is given an undirected graph G on n vertices and two vertices s ant t. The algorithm should run in space $O(\log n)$ and output an answer in $\{0,1\}$ such that:

   - If the graph G does not have a path from s to t then the algorithm outputs 0.
   - If the graph G has a path from s to t that is of length at most $100 \log n$ and every vertex v in the graph has degree at most 100 then the algorithm outputs 1.

3. (22) Show that: $PSPACE \subseteq P/poly \Rightarrow PSPACE = MA$. (Hint: it is easy to see that $MA \subseteq PSPACE$. For the other direction note that $PSPACE = IP$ and that we noted that the prover strategy in an IP protocol can be computed in $PSPACE$.)

4. Consider the language:

   $$L = \{M, x, t : \text{deterministic TM } M \text{ halts on } x \text{ after at most } t \text{ steps}\}$$

   (a) (5) Show that $L \in EXP$.

   (b) (17) Show that $L \notin P$.

5. (25) (zero sided error versus expected polynomial time)

Definition 3. *A language L has a zero-sided error poly-time probabilistic algorithm if there exists a probabilistic poly-time algorithm A which gives outputs in $\{0, 1, ?\}$ and satisfies:*

- *For every input x and coin-toss y if $A(x, y) \neq ?$ then $A(x, y) = 1_L(x)$.*
- *For every input x, $\Pr_y[A(x, y) = ?] \leq 1/3$.*

Definition 4. *A language L is in ZPP if there exists a probabilistic algorithm A such that for every input x and every coin toss y, $A(x, y) = 1_L(x)$ and there exists a polynomial p such that for every input x the expected running time of A on x is bounded by $p(|x|)$.*

Show that a language L is in ZPP if and only if it has a zero-sided error poly-time probabilistic algorithm.

6. ($P/\log$). Recall that $P/\log$ is the class of all languages accepted by nonuniform Turing machines that receive advice of logarithmic length. More precisely, a language $L \in P/\log$ if there exist a polynomial time Turing machine and a sequence $\{\alpha_n\}_{n=1}^{\infty}$ of strings such that for every n, $|\alpha_n| = \log n$ and for every input x, $M(x, \alpha_{|x|}) = 1_L(x)$.

   (a) (5) Show that $P/\log \neq P$.

   (b) (20) Show that if $NP \subseteq P/\log$ then $NP = P$.

7. ($NTIME(n) \neq P$) We say that a class C of languages allows unpadding by a function t if whenever $L_t \in C$ then $L \in C$. (Here $L_t = \left\{1^{t(|x|)}0 \circ x : x \in L\right\}$ as in exercise 2).

   (a) (10) Show that P allows unpadding by any polynomial $t(n)$.

   (b) (15) Show that $NTIME(n)$ does not allow unpadding by $t(n) = n^5$.

   (c) Conclude that $NTIME(n) \neq P$. Did we just prove that $SAT \notin P$? (No need to answer).

8. (30) (parallel repetition of AM).

   Show that for every polynomial $p(n)$, $AM_{1/3}^{2/3} = AM_{1/2^{p(n)}}^{1-1/2^{p(n)}}$.

9. (AM and perfect completeness).

   (a) (25) Show that $AM_{1/3}^{2/3} = AM_{1/3}^{1}$. (Hint: use the techniques of the proof that $BPP \subseteq \Sigma_2^P$).

   (b) (5) Conclude that $AM \subseteq \Pi_2^P$.

10. (MA and perfect completeness).

    (a) (10) Show that for every polynomial $p(n)$, $MA_{1/3}^{2/3} = MA_{1/2^{p(n)}}^{1-1/2^{p(n)}}$.

    (b) (15) Show that $MA_{1/3}^{2/3} = MA_{1/3}^{1}$.

11. (Finding hard instances for NP)

(a) (15) Assume that for every polynomial time TM A, and for every sufficiently large n, there exists a satisfiable formula ϕ such that $A(\phi)$ fails to produce a satisfying assignment. (Think of A as an algorithm suggested by an adversary who claims to solve satisfiabiliy). Show that for every polynomial time TM A, there exists a polynomial time TM B which given input 1^n, produces a satisfiable formula ϕ of length at least n, such that $A(\phi)$ fails to to produce a satisfying assignment. (Hint: consider the statement "there exists a satisfiable formula on which A fails".)

(b) (20) (Show that if $NP \neq P$ then for every for every polynomial time TM A that answers yes/no (again, think of A as an algorithm trying to solve satisfiability), there exists a polynomial time TM B which given input 1^n, produces a constant number of formulas all of length at least n, such that for infinitely many n, one of the produced formulas is one on which A fails to answer correctly whether the formula is satisfiable.