Take Home Final

due June 2015

Do at least (4, 6) 5 (or thereabouts) problems. There are mix of easy (3, 4, 11) and harder problems, and all are fun.

1. Show that the GEN problem is $P$-complete.

2. Show that GEN is $NL$-complete if the operation is restricted to being associative.

3. Prove that $ZPP = RP \cap coRP$. (This is problem 1 on page 42 of the Trevisan notes. See chapter 5 there or the Papadimitriou text for definitions of $ZPP$ and $RP$. Also, it’s pretty easy.)

4. Prove that if $NP \subseteq BPP$ then $NP \subset RP$. (This is problem 2 on page 42 of the Trevisan notes, and is very easy.)

5. Is $N \circ \Sigma^P_2 = N \circ \Pi^P_2$? (See below for the $N \circ$ and $BP \circ$ notation for this and the next two problems.)

6. Argue that $NBPP \subseteq BPNP$.

7. Show that $NP^{BPP} \subseteq NBPP \subseteq ZPP^{NP}$.

8. Let $H$ be the language consisting of tuples $< x, 1^{2^n}, 1^s >$ so that $x$ is a boolean string of length less than $2^n$ that is the prefix of some truth table of length $2^n$ whose corresponding boolean function cannot be computed by any circuit of size $s$. Show that $H$ is in $\Sigma^P_2 = N^{P^{NP}}$. (Very hard to parse, easy to do.)

9. A strong nondeterministic TM is one that has three possible halt states: “yes”, “no”, or “maybe”. We say such a machine decides $L$ in polynomial time if all computations run in polynomial time, and if the following holds: if $x \in L$, then all computations end up with “yes” or “maybe”, and at least one ends up “yes”. If $x \notin L$, then all computations end up in “no” or “maybe”, but at least one ends up “no”. Prove that $L$ is decided by a strong nondeterministic TM in polynomial time iff $L \in NP \cap coNP$.

10. In fact, we say that $A$ is strong-nondeterministically reducible to $B$, $A \leq^{SN} B$, if $A \in NP^B \cap coNP^B$. Show that the set of $\leq^{SN}$-complete sets for NP is precisely $H^P_1 = \{ A | \Sigma^P_2 \subseteq NP^A \}$. (See below for comments on the high and low sets.)

11. Show that if every NP-hard problem is also PSPACE-hard, then NP=PSPACE. (This is pretty easy.)

Definitions and comments:
• The $GEN$ problem is as follows: you are given a set $W$, an operation $\circ$ on $W$, a subset $V \subseteq W$, and an element $x \in W$. Determine if $x$ is in the closure of $V$ under $\circ$ (that is, whether $x$ can be generated by applying the operation $\circ$ to some elements of $V$ in some order).

• We define quantifier operators that construct one complexity class from another. Let $C$ be a complexity class.
  
  – The class $N \circ C$ is defined to be the class of sets $A$ which can be characterized as
    \[
    A = \{ x \mid \exists y \ (|y| \leq |x|^k) \langle x, y \rangle \in B \}
    \]
    for some $B \in C$ and integer $k$.
  
  – $N \circ C$ is not NC (aka “Nick’s Class”). However we can define $NP = N \circ P$. Also now define $MA = NBPP = N \circ BPP$.
  
  – Next we move to $BP \circ C$, which consists of those sets $A$ that can be described by a $B \in C$ and an integer $k$ in the following way: on input $x$ pick a random $y$ $(|y| \leq |x|^k)$, and test whether $\langle x, y \rangle \in B$. If $x \in A$, then this probability should be at least $\frac{3}{4}$; if $x \notin A$, then it should be at most $\frac{1}{4}$.
  
  – We have seen $BPP = BP \circ P$. Also of interest is $BPNP = BP \circ NP$, which is sometimes called $AM$.

• Schöning introduced the concept of lowness and highness for the polynomial hierarchy. A set $A$ is in $L^P_k$ iff $\Sigma^P_k \subseteq \Sigma^P_k$ and is in $H^P_k$ iff $\Sigma^P_k \supseteq \Sigma^P_{k+1}$. It is not too hard to see that $P = L^P_0 \subseteq L^P_1 \subseteq L^P_2 \subseteq \cdots$ and the $\leq^P$-complete sets for $NP = H^P_0 \supseteq H^P_1 \supseteq H^P_2 \supseteq \cdots$. On HW2 you showed that $L^P_1 = NP \cap coNP$. 