Theory Qual, Spring 2011
Complexity

February 18, 2011

Duration: 3 hours
Number of problems: 5

1 Languages are closed under homomorphism.

Recall the following concepts.

- A (alphabet) homomorphism from Σ to Δ is a function $h : \Sigma^* \to \Delta^*$ such that $h(uv) = h(u)h(v)$.
- Given a homomorphism h from Σ to Δ, and a language $L \subseteq \Delta^*$, the *inverse homomorphic image* of L is $h^{-1}(L) = \{w \in \Sigma^* \mid h(w) \in L\}$.
- Let $\mathcal{C}$ be a collection of languages. We say $\mathcal{C}$ is closed under *regular intersection* if and only if for every $L \in \mathcal{C}$ and any regular language R, $L \cap R \in \mathcal{C}$. $\mathcal{C}$ is said to be closed under homomorphism (inverse homomorphism) if and only if for every $L \in \mathcal{C}$ and any homomorphism h, $h(L) \in \mathcal{C}$ ($h^{-1}(L) \in \mathcal{C}$).

A finite transducer is a machine model that computes functions in "constant space". Formally, $M = (Q, \Sigma, \Delta, q_0, \delta, \lambda)$ is a finite transducer, where Q is a finite set of states, Σ is the input alphabet, Δ is the output alphabet, $q_0 \in Q$ is the initial function, and $\delta : Q \times \Sigma \to Q$ is the transition function. The output function is $\lambda : Q \times \Sigma \to \Delta^*$. On a word $w = w_1 w_2 \cdots w_n$ if the unique execution of M is $q_0 = r_0 \xrightarrow{w_1} r_1 \xrightarrow{w_2} r_2 \cdots \xrightarrow{w_n} r_n$, then the output of M on w, $M(w)$, is the word $\lambda(r_0, w_1)\lambda(r_1, w_2) \cdots \lambda(r_i, w_{i+1}) \cdots \lambda(r_{n-1}, w_n)$. For a language $L \subseteq \Sigma^*$, $M(L) = \{M(w) \mid w \in L\}$. Finally, recall that a collection of languages $\mathcal{C}$ is said to be *closed under finite transductions* if and only if for every $L \in \mathcal{C}$ and any transducer M, $M(L) \in \mathcal{C}$.

Prove that $\mathcal{C}$ is closed under finite transductions iff $\mathcal{C}$ is closed under homomorphisms, inverse homomorphisms, and regular intersection.

2 Polynomial reductions

Recall that a language L_1 is said to be reducible in polynomial time to L_2 if there is a reduction f from L_1 to L_2 that can be computed in polynomial time; we will denote this by $L_1 \leq_m^P L_2$. We will say $L_1 \equiv_m^P L_2$ if $L_1 \leq_m^P L_2$ and $L_2 \leq_m^P L_1$. An *(polynomial) m-degree* is an equivalence class of $\equiv_m^P$. Since reductions compose, and every language is reducible to itself, $\leq_m^P$ is a partial order on the m-degrees. Prove that the m-degrees form a join semi-lattice with respect to $\leq_m^P$, i.e., if $\mathcal{C}_1$ and $\mathcal{C}_2$ are m-degrees then there is a unique m-degree $\mathcal{C}$ such that

Upper Bound $\mathcal{C}_1 \leq_m^P \mathcal{C}$ and $\mathcal{C}_1 \leq_m^P \mathcal{C}$

Least Upper Bound If $\mathcal{C}'$ is an upper bound of $\mathcal{C}_1$ and $\mathcal{C}_2$ then $\mathcal{C} \leq_m^P \mathcal{C}'$

3 Super-duper NP-Completeness.

Recall that NP-completeness is defined as follows. L is NP-complete if (a) $L \in NP$, and (b) for every $L' \in NP$, there is polynomial p and function f such that $f(x)$ is computable in time $p(|x|)$ and $x \in L \leftrightarrow f(x) \in L'$. Let us define a new notion of NP-super-complete by permuting one of the quantifiers as follows. L is NP-super-complete if (a) $L \in NP$, and (b) there is polynomial p, such that for every $L' \in NP$ there is a function f with the property that $f(x)$ is computable in time $p(|x|)$ and $x \in L \leftrightarrow f(x) \in L'$. Prove that there are no NP-super-complete languages.

4 Circuit of constant depth.

Recall that a uniform $\mathbf{NC}^0$ circuit family consists of boolean circuits of constant depth and constant fan-in, such that the circuits in the family can be generated by a logarithmic space Turing Machine (logarithmic in the size of the input of circuit generated). Note that the output bit of an $\mathbf{NC}^0$ circuit can depend only on a constant number of bits of the input.

We consider $\mathbf{NC}^0$ circuits which can output multiple bits. We shall say that such a circuit *accepts* its input if all the output bits are 1, and else rejects its input. A multi-bit output $\mathbf{NC}^0$ circuit family is said to decide a language if for every binary string x, the circuit of input-size $|x|$ accepts x if and only if x is in the language. We define a class of languages $\mathbf{NC}^0_{\text{AND}}$ to consist of languages that are decided (in the above sense) by uniform, multi-bit output $\mathbf{NC}^0$ circuit families.

(A) (easy.) Show that $\mathbf{NC}^0_{\text{AND}} \subseteq \mathbf{P}$.

(B) Recall that **NP** is the class of languages L for which there is a language $R \in \mathbf{P}$ such that $L = \{x | \exists w, |w| = \text{poly}(|x|), (x, w) \in R\}$. Show that instead of $R \in \mathbf{P}$, if we use $R \in \mathbf{NC}^0_{\text{AND}}$, the class defined is still **NP**.

[*Hint: Can you think of extra information to be provided along with the witness to enable lower complexity for verification?*]

5 No false witness.

This problem deals with *counting* the number of witnesses for an input in an **NP** language L, given an oracle for all **NP** languages (or say for SAT).

You can assume that for any k, m of your choice, you have a family of hash functions $\mathcal{H}$ such that for $h \in \mathcal{H}$, $h : \{0,1\}^k \to \{0,1\}^m$ is computable in time $\text{poly}(k, m)$ and has the following property: given any sufficiently large subset $S \subset \{01\}^k$, for at least half of the functions $h \in \mathcal{H}$, the partition of the domain induced by h "shatters" S into small subsets. More precisely, there exist constants c and ϵ (independent of k, m) such that for all $S \subseteq \{0,1\}^k$, $|S| \geq ck$, for at least $\frac{1}{2}$ fraction of $h \in \mathcal{H}$,

$$\forall y \in \{0,1\}^m, \quad |h^{-1}(y) \cap S| \leq (1+\epsilon)|S|/2^m.$$

Let $L = \{x | \exists w, |w| = \ell(|x|), (x, w) \in R\}$. for a language $R \in \mathbf{P}$, and ℓ a polynomial. For each x, we define $\sharp R(x) := |\{w \mid |w| = \ell(|x|), (x, w) \in R\}$.

Give a polynomial-time oracle TM that, given access to a SAToracle, on input x finds out if $\sharp R(x) \leq c\ell(|x|)$ or not, and:

(A) if $\sharp R(x) \leq c\ell(|x|)$, returns the list of all w such that $(x, w) \in R$.

(B) else, with probability $\geq \frac{3}{4}$ finds an approximation $\sharp R(x)(1 \pm \delta)$ for some small constant δ (depending on ϵ).

[*Hint: You can use the hash function for the second part. Given an h, can you use the* **NP** *oracle to check if it indeed shatters the set of witnesses?*]