Thoughts on Covering U_n

Eric Allender, Samuel Hopkins

July 18, 2011

This document contains some thoughts about the a potential new proof technique.

1 Notation

Let $A \in \mathsf{REC}$ with $A \leq_{tt}^{f(n)} R_C$ and let this be witnessed by the reduction $M(x)$ which generates query set $Q(x)$ (where $|Q(x)| \leq f(n)$ for $x \in \Sigma^n$). Let $s_n = a \log f(n) + b \log\log f(n) + c$ for some appropriate $a, b, c \in \mathbb{N}$. Let $Q'(x) = Q(x) \cap \{0,1\}^{\leq s_n}$.

By an *answer vector* to some $Q'(x)$ we just mean a function $v : Q'(x) \to \{0,1\}$ which conjectures answers to the queries in $Q'(x)$. We write $M^v(x)$ to denote the result of the reduction of running the reduction M on x using answers from v rather than from R_C, and we write G_x (for "good for x") to denote the set of answer vectors v for $Q'(x)$ that are such that, for any y with $Q'(y) = Q'(x)$, we have $M^v(y) = A(y)$ (we conflate A and its characteristic function). We say that such a vector is good for x or just good if the context makes clear the rest.

Let v_0^x be the answer vector that sends $Q'(x) \cap R_C$ and nothing else to 1. For obvious reasons we call v_0^x the *correct vector for x* or just *correct*.

We denote by $\mathcal{C}$ the set of answer vectors *compatible with* R_C. A vector v is compatible with R_C if $v(x) = 1$ implies that $x \in R_C$.

1.1 Passing to New s_n's

Fix some constants a, b, c, so that s_n is now fixed. Fix another set of constants $\overline{a}, \overline{b}, \overline{c}$ and let them fix $\overline{s_n}$, and let them be chosen such that $\overline{s_n} > s_n$. Let $Q'(x) = Q(x) \cap \{0,1\}^{<s_n}$ and let $\overline{Q'}(x) = Q(x) \cap \{0,1\}^{<\overline{s_n}}$. Let G_x be the good vectors for x with respect to $Q'(x)$ and let $\overline{G_x}$ be the good vectors for x with respect to $\overline{Q'(x)}$.

It's obvious that $v \in G_x$ iff there is $\overline{v} \in \overline{G_x}$ such that $\overline{v}$ agrees with v on all queries in $Q'(x)$ and answers negatively to all queries in $\overline{Q'}(x) \setminus Q'(x)$.

1.2 Covering U_n

We keep the notation of the preceding section. Let

$$U_n = \bigcup_{|x|=n} Q'(x).$$

The set U_n is the "universe" of all short queries (where "short" means "short with respect to s_n") that could appear on an input of length n.

Choose x_0 so as to maximize $|Q'(x_0)|$ and x_1 so as to maximize $|Q'(x_1) \setminus Q'(x_0)|$, and so forth. Continue the sequence $x_0, x_1, \cdots, x_m$ until it has no good vector.

Suppose first that U_n is not covered by $Q'(x_0), Q'(x_1), \cdots, Q'(x_m)$. Letting $S = x_0, x_1, \cdots, x_m$, we have that

$$|S| \leq |U_n| \leq 2^{s_n}.$$

Now, we know that there is some $x_i \in S$ such that $v_0^{x_i} \notin G_{x_i}$. It costs at most s_n bits to point to x_i. But we can get a better bound. We know that for sufficiently-long x, as long as a is big enough, the first, say, $f(n)$ elements x_j of S are such that $v_0^{x_j} \notin G_{x_j}$. So our bound on the number of bits to point to x_i is actually $s_n - \log f(n)$. (Is this true?? There will be some $\mathcal{O}(1)$ term here that we haven't dealt with.)

Now there are two things to try. One is to see if we can get the size of the pointer to be low enough that $v_0^{x_i} \in G_{x_i}$, which would be a contradiction. This would require a lot of coefficient accounting. The other is to pass to some larger $\overline{s_n}$. We would then have $v_0^{x_i} \in \overline{G_{x_i}}$, which means that there is some actually-random query in $Q(x_i)$ with length greater than s_n but less than $\overline{s_n}$. It would cost $\log f(n)$ bits to get a pointer in to it—can we free up $\log f(n)$ bits?

Things to prove: that we can acutually assume that for the first and last $f(n)$ elements of S we have $v_0^{x_j} \in G_{x_j}$; that when we do all the concatenation of pieces of information, etc., that we get the right bounds.

2 Beginnings of an Argument

Let $s_1(n) = 3\log f(n) + 2\log\log f(n) + c$ for some appropriately-chosen c. Let

$$U_1(n) = \bigcup_{|x|=n} Q_1'(x)$$

where $Q_1'(x)$ is Q' with respect to s_1. Then start trying to cover $U_1(n)$ greedily with some sequence of strings $x_0, x_1, \cdots, x_m$, at each step maximizing $|Q_1'(x_i) \setminus \bigcup_{j<i} Q_1'(x_j)|$. Stop when the resulting sequence S no longer has any good vectors. Now we split into cases:

2.1 Case One

Suppose first that we have not covered $U_1(n)$ when the sequence S terminates. Then $|S| \leq |U_1(n)|$. It therefore costs at most $s_1(n)$ bits to get a pointer in to S.

Proposition. For x_i with $m - f(n) < i \leq m$ it is the case that $v_0^{x_i} \in G_1(x_i)$. The same holds for $0 \leq i < f(n)$.

Proof. We will prove the proposition for $m - f(n) < i \leq m$—the other argument is symmetric. For each of those x_i, we have $C(x_i) \leq \log f(n) + d$ where d is some constant encoding the Turing Machine that executes the covering algorithm. The number d will be constant with respect to n, so for large n we have that there exist some a, b, c fixing $s_2(n)$ such that $v_0^{x_i} \in G_2(x_i)$ (where $G_2(x)$ is the set of good vectors for x with respect to s_2). WLOG, $s_2(n) > s_1(n)$ for all (large) n.

Now, suppose that there are arbitrarily-large n where there is some x_i with $m - f(n) < i \leq m$ such that $v_0^{x_i} \notin G_1(x_i)$. Since $v_0^{x_i}$ *is* in $G_2(x_i)$, there must be some truly random string $q \in Q(x)$ with $s_1 < |q| < s_2$. It costs $\log f(n)$ bits to get a pointer to x_i and another $\log f(n)$ bits to point into $Q(x_i)$ at the random string, so $C(q) \leq 2 \log f(n) + \log \log f(n) + e$ for some constant e. But $|q| > 3 \log f(n)$. Since q is truly random, we have a contradiction for large n. This concludes the proof of the proposition.

□

We return now to the main line of argument for this first case. We know that there is some $x_j \in S$ such that $v_0^{x_j} \notin G_1(x_j)$. It cannot be the case (for large n) that $m - f(n) < i \leq m$ or that $0 \leq i < f(n)$. It therefore costs at most $s_1(n) - 2 \log f(n)$ bits to point to x_j. Now we will pull the same argument again. There are some coefficients a, b, c fixing $s_2(n)$ such that $C(x_j)$ is sufficiently low that $v_0^{x_j} \in G_2(x_j)$. Therefore there is some truly random query $q \in Q(x_j)$ with $|q| > s_1(n)$. It costs $s_1(n) - 2 \log f(n)$ bits to point to x_j and $\log f(n)$ bits to point to q in $Q(x_j)$. Therefore (modulo some constant term), $C(x) \leq s_1(n) - \log f(n)$. But this is a contradiction for large n. So case one is impossible.

2.2 Case Two

Suppose now that we have covered $U_1(n)$ when the sequence S terminates. Then $|S| \geq |U_1(n)|$.