

Solution Set 6

Posted: May 28

Chris Umans

1. (a) We observe that the largest possible set shattered by a collection of 2^m subsets is m , since a set of size $m + 1$ has more than 2^m distinct subsets. The VC dimension of a collection of subsets succinctly encoded by a circuit C can therefore be at most $|C|$, since C can encode at most $2^{|C|}$ subsets. Thus we can express VC-DIMENSION as follows:

$$\{(C, k) : \exists X \forall X' \subseteq X \exists i [|X| \geq k \text{ and } \forall y \in X C(i, y) = 1 \Leftrightarrow y \in X']\}$$

Notice that $|X|$, $|X'|$, and $|i|$ are all bounded by $|C|$ (using the observation above), and that the expression in the square brackets is computable in $\text{poly}(|C|)$ time. Thus VC-DIMENSION is in Σ_3^p .

- (b) Let $\phi(a, b, c)$ be an instance of QSAT₃ (so we are interested in whether $\exists a \forall b \exists c \phi(a, b, c)$). We may assume by adding dummy variables if necessary that $|a| = |b| = |c| = n$. As suggested our universe is $U = \{0, 1\}^n \times \{1, 2, 3, \dots, n\}$. We identify n -bit strings with subsets of $\{1, 2, 3, \dots, n\}$, and define our collection \mathcal{S} of sets to be the sets

$$S_{a,b,c} = \begin{cases} \{a\} \times b & \text{if } \phi(a, b, c) = 1 \\ \emptyset & \text{otherwise} \end{cases}$$

for all a, b, c .

There is a small circuit C that succinctly encodes this collection of sets – given an element $x = (a', k) \in U$ and a set name (a, b, c) , determining whether $x \in S_{a,b,c}$ requires only that we check if $\phi(a, b, c) = 1$ (if it is not, then the set is the empty set and clearly $x \notin S_{a,b,c}$) and then check if $x \in \{a\} \times b$ (i.e., check whether $a' = a$ and $b_k = 1$). Our instance of VC-DIMENSION is (C, n) .

If ϕ is a positive instance, i.e., $\exists a \forall b \exists c \phi(a, b, c) = 1$, then the set $U_a = \{a\} \times \{1, 2, 3, \dots, n\}$ of size n is shattered, because \mathcal{S} contains sets of the form $\{a\} \times b$ for all b . Thus the VC dimension of \mathcal{S} is at least n .

Conversely, if the VC dimension of \mathcal{S} is at least n , then there is a set X of size n that is shattered by \mathcal{S} . We observe that X cannot contain elements of two different subsets U_a and $U_{a'}$ because then the set consisting of these two elements cannot be expressed as the intersection of X with some set in \mathcal{S} (all of our sets are subsets of some U_a). We conclude that $X \subseteq U_a$ for some a , and the fact that it is shattered implies that sets of the form $\{a\} \times b$ for all b must be present in \mathcal{S} . This implies that $\forall b \exists c \phi(a, b, c)$, so we have a positive instance.

We have shown that (C, n) is a positive instance of VC-DIMENSION iff ϕ is a positive instance of QSAT₃, as required.

2. (a) Let C_1, C_2 be two circuits. The circuit $C(x, y) = C_1(x) \wedge C_2(y)$ has a number of satisfying assignments equal to the product of the number of satisfying assignments of C_1 and the number of satisfying assignment of C_2 . Observe that the size of C is at most $|C_1| + |C_2| + O(1)$

To handle the sum, we first define $C'_1(x, y)$ to be the circuit that outputs 1 iff $C_1(x)$ outputs 1 and y is the all-zeros string, and $C'_2(x, y)$ to be the circuit that outputs 1 iff $C_2(y)$ outputs 1 and x is the all-zeros string. Clearly the number of satisfying assignments of C'_1 is the same as the number of satisfying assignments of C_1 and similarly for C'_2 and C_2 . This manipulation ensures that both circuits are defined over the same set of inputs. Now, the circuit $C(z, x, y) = (z \wedge C'_1(x, y)) \vee (\neg z \wedge C'_2(x, y))$ (where z is a single fresh Boolean variable) has a number of satisfying assignments equal to the sum of the number of satisfying assignments of C'_1 and the number of satisfying assignment of C'_2 . Observe that the size of C is at most $|C_1| + |C_2| + O(n)$, where n is the number of variables of C_1 and C_2 .

Let B be the number of satisfying assignments of C . Given the polynomial $g = \sum_i a_i t^i$, we can produce circuits C_i with a number of satisfying assignments equal to B^i by applying the “product” transformation to C with itself i times. By the above observation $|C_i| \leq \deg(g)|C| + O(\deg(g))$.

We can easily produce a circuit D_i that has exactly a_i satisfying assignments as follows: D_i has $\lceil \log_2 a_i \rceil$ variables, it treats its input as a nonnegative integer, and outputs 1 iff that integer is less than a_i . Thus circuit D_i has size $O(\log a_i)$. We now produce a circuit C'_i with a number of satisfying assignments equal to $a_i B^i$, by applying the “product” transformation to the circuits D_i and C_i . The resulting circuit has size at most $|C_i| + O(\log a_i)$.

Finally, we apply the “sum” transformation $\deg(g) - 1$ times to produce a circuit C' from the C'_i with a number of satisfying assignments equal to $\sum_i a_i B^i = g(B)$. If $A = \max_i a_i$, we have

$$|C'| \leq O\left(\sum_i |C'_i|\right) \leq \deg(g) \cdot O(\deg(g)|C| + O(\log A))$$

which is polynomial in $|C|$ and the size of polynomial g when written in the natural way as a vector of coefficients (each of which takes at most A bits to write down).

- (b) Let's check the property of g_0 . We have:

$$g_0(Y) = Y^2(3 - 2Y)$$

and plugging in a multiple of 2^{2^i} for Y we see that the result is a multiple of $(2^{2^i})^2 = 2^{2^{i+1}}$. This verifies the first property. Also,

$$g_0(Y + 1) = 3(y^2 + 2Y + 1) - 2(Y^3 + 3Y^3 + 3Y + 1) = -2Y^3 - 3Y^2 + 1$$

Plugging in any multiple of 2^{2^i} for Y into this shifted polynomial we see that the result is 1 plus a multiple of $(2^{2^i})^2 = 2^{2^{i+1}}$, which verifies the second property.

Let $m = 2^k$ for a positive integer k . Then by composing g_0 with itself k times, we produce the required polynomial g . The composed polynomial has degree $3^k = \text{poly}(m)$, and nonnegative integer coefficients of magnitude at most $3^{(3^k)} = \exp(\text{poly}(m))$ so the entire

polynomial can be written down is space $\text{poly}(m)$. Actually performing the composition just requires multiplying out the terms which can easily be done in time $\text{poly}(m)$.

- (c) We know from the last problem set that the PH is contained in $BPP^{\oplus P}$. Fix a language L in $BPP^{\oplus P}$. We first observe that we can have the BPP machine flip all of its coins first (writing them down) and then proceed with a deterministic computation whose input is the original input plus the random coins. In other words L can be decided by a BPP oracle TM that makes a *single* oracle query to a $P^{\oplus P}$ oracle, and enters q_{accept} if the answer is “yes” and q_{reject} if the answer is “no.” By Problem 2(d) on the last problem set $P^{\oplus P} \subseteq (\oplus P)^{\oplus P} \subseteq \oplus P$, so this oracle can be replaced with an $\oplus P$ oracle. So now we have a $BPP^{\oplus P}$ machine with the special structure suggested by the hint, and let r be the number of coins it tosses. Let M be the nondeterministic TM associated with the $\oplus P$ oracle language, and let C_y denote the CIRCUIT SAT instance obtained from M on input y . On a given computation path where $w \in \{0, 1\}^r$ are the random coins tossed by the BPP machine, resulting in oracle query $y = f(w)$, the $BPP^{\oplus P}$ machine enters q_{accept} iff the number of satisfying assignments to C_y is odd, and q_{reject} otherwise. Put another way, it enters q_{accept} if the number of satisfying assignments is $1 \pmod 2$ and q_{reject} if the number of satisfying assignments is $0 \pmod 2$.

By applying parts (a) and (b), we can efficiently produce from C_y a circuit C'_y for which the number of satisfying assignments to C'_y is either 0 or 1 *modulo* $B = 2^{r+1}$. Where does this get us? In the case of an input $x \in L$, there are at least $(2/3)2^r$ paths of the BPP machine that produce a circuit C'_y with a number of satisfying assignments that is $1 \pmod B$ and the others produce a circuit C'_y with a number of satisfying assignments that is $0 \pmod B$. In the case of an input $x \notin L$, there are at most $(1/3)2^r$ paths of the BPP machine that produce a circuit C'_y with a number of satisfying assignments that is $1 \pmod B$ and the others produce a circuit C'_y with a number of satisfying assignments that is $0 \pmod B$.

So, given input x , if we *count* the number of (w, z) pairs (where w is a sequence of r random coins tossed by the BPP machine) for which $C'_{f(w)}(z) = 1$, this number *modulo* B will be equivalent to something between $(2/3)2^r$ and 2^r if $x \in L$ and something between 0 and $(1/3)2^r$ if $x \notin L$. Thus we can decide L in $P^{\#P}$, since we can recognize the set of (w, z) pairs for which $C'_{f(w)}(z) = 1$ in polynomial time (so getting a raw count can be done in $\#P$, and then the P machine only needs to take the result modulo B).

3. (a) We describe R' separately for strings x of each length. Consider strings x of length m and assume $|z| = |x|^c$. Set $k = m^{3c}$ and $n = k^2$, and let $E : \{0, 1\}^n \times \{0, 1\}^t \rightarrow \{0, 1\}^{m^c}$ be a (k, ϵ) extractor with $\epsilon < 1/6$ and $t = O(\log n)$. Define the language \widehat{R} to be those triples (x, y, \widehat{z}) for which $(x, y, E(\widehat{z}, w)) \in R$ for more than half of the $w \in \{0, 1\}^t$. Since R is in \mathbf{P} and $t = O(\log n)$, \widehat{R} is also in \mathbf{P} . We now claim that

- If $x \in L$, then there exists y for which

$$|\{\widehat{z} : (x, y, \widehat{z}) \notin \widehat{R}\}| \leq 2^{n^{1/2}}.$$

To prove this, take y to be the y for which $\Pr_z[(x, y, z) \in R] \geq 2/3$ (guaranteed by the definition), and call a \widehat{z} in the above set “bad.” For \widehat{z} to be bad, it must be that

$$|\Pr_z[(x, y, z) \in R] - \Pr_w[(x, y, E(\widehat{z}, w)) \in R]| > 1/6,$$

(since the left probability is at least $2/3$, and the right one must be less than $1/2$ for bad \hat{z}). Thus there must be fewer than $2^k = 2^{n^{1/2}}$ bad \hat{z} (because the set of bad \hat{z} comprise a source with minentropy k on which the extractor fails).

- If $x \notin L$, then for all y

$$|\{\hat{z} : (x, y, \hat{z}) \in \widehat{R}\}| \leq 2^{n^{1/2}}.$$

To prove this, fix a y and call a \hat{z} in the above set “bad.” For \hat{z} to be bad, it must be that

$$|\Pr_z[(x, y, z) \in R] - \Pr_w[(x, y, E(\hat{z}, w)) \in R]| > 1/6,$$

(since the left probability is at most $1/3$, and the right one must be at least $1/2$ for bad \hat{z}). Thus there must be fewer than $2^k = 2^{n^{1/2}}$ bad \hat{z} for the same reason as above.

Now we can define R' . The idea is to split \hat{z} into two equal-length halves: $\hat{z} = (\hat{z}_1, \hat{z}_2)$. Then we define R' to be those $(x, y' = (y, \hat{z}_1), z' = \hat{z}_2)$ for which $(x, y, \hat{z}) \in \widehat{R}$. Let's check that this satisfies the requirements. If $x \in L$, then there exists a y and a \hat{z}_1 for which for all \hat{z}_2 , $(x, y, \hat{z}) \in \widehat{R}$ (if not, then there would be at least $2^{n/2} > 2^{n^{1/2}}$ distinct \hat{z} for which $(x, y, \hat{z}) \notin \widehat{R}$, contradicting our analysis above). And, if $x \notin L$, then we claim that for all y and all \hat{z}_1 , $\Pr_{\hat{z}_2}[(x, y, \hat{z}) \in \widehat{R}] < 1/3$. If not, then for some y there would be at least $(2/3)2^{n/2} > 2^{n^{1/2}}$ distinct \hat{z} for which $(x, y, \hat{z}) \in \widehat{R}$, contradicting our analysis above.

- (b) As in part (a), we describe R' separately for strings x of each length. Consider strings x of length m and assume $|y| = |x|^c$. Set $k = m^{3c}$ and $n = k^2$, and let $E : \{0, 1\}^n \times \{0, 1\}^t \rightarrow \{0, 1\}^{m^c}$ be a (k, ϵ) extractor with $\epsilon < 1/6$ and $t = O(\log n)$. Define the language \widehat{R} to be those triples $(x, \hat{y}, (z_w)_{w \in \{0, 1\}^t})$ for which $(x, E(\hat{y}, w), z_w) \in R$ for more than half of the $w \in \{0, 1\}^t$. Since R is in \mathbf{P} and $t = O(\log n)$, \widehat{R} is also in \mathbf{P} . We now claim that

- If $x \in L$, then we claim

$$|\{\hat{y} | \forall (z_w)_{w \in \{0, 1\}^t} (x, \hat{y}, (z_w)_{w \in \{0, 1\}^t}) \notin \widehat{R}\}| \leq 2^{n^{1/2}}.$$

Call a \hat{y} in the above set “bad.” For \hat{y} to be bad, it must be that

$$|\Pr_y[\exists z (x, y, z) \in R] - \Pr_w[\exists z (x, E(\hat{y}, w), z) \in R]| > 1/6,$$

(since the left probability is at least $2/3$, and the right one must be less than $1/2$ for bad \hat{y}). Thus there must be fewer than $2^k = 2^{n^{1/2}}$ bad \hat{y} (because the set of bad \hat{y} comprise a source with minentropy k on which the extractor fails).

- If $x \notin L$, then we claim

$$|\{\hat{y} : \exists (z_w)_{w \in \{0, 1\}^t} \text{ for which } (x, \hat{y}, (z_w)_{w \in \{0, 1\}^t}) \in \widehat{R}\}| \leq 2^{n^{1/2}}.$$

Call a \hat{y} in the above set “bad.” For \hat{y} to be bad, it must be that

$$|\Pr_y[\exists z (x, y, z) \in R] - \Pr_w[\exists z (x, E(\hat{y}, w), z) \in R]| > 1/6,$$

(since the left probability is at most $1/3$, and the right one must be at least $1/2$ for bad \hat{y}). Thus there must be fewer than $2^k = 2^{n^{1/2}}$ bad \hat{y} for the same reasons as above.

Now we can define R' . Similar to before, the idea is to split \hat{y} into two equal-length halves: $\hat{y} = (\hat{y}_1, \hat{y}_2)$. Then we define R' to be those $(x, y' = \hat{y}_1, z' = (\hat{y}_2, (z_w)_{w \in \{0,1\}^t}))$ for which $(x, \hat{y}, (z_w)_{w \in \{0,1\}^t}) \in \hat{R}$. Let's check that this satisfies the requirements. If $x \in L$, then for all \hat{y}_1 , there exist $\hat{y}_2, (z_w)_{w \in \{0,1\}^t}$ for which $(x, \hat{y}, (z_w)_{w \in \{0,1\}^t}) \in \hat{R}$ (if not, then there would be at least $2^{n/2} > 2^{n^{1/2}}$ distinct \hat{y} for which

$$\forall (z_w)_{w \in \{0,1\}^t} (x, \hat{y}, (z_w)_{w \in \{0,1\}^t}) \notin \hat{R},$$

contradicting out analysis above). And, if $x \notin L$, then we claim that

$$\Pr_{\hat{y}_1}[\exists \hat{y}_2, (z_w)_{w \in \{0,1\}^t} \text{ for which } (x, \hat{y}, (z_w)_{w \in \{0,1\}^t}) \in \hat{R}] \leq 1/3.$$

If not, then there would be at least $(1/3)2^{n/2} > 2^{n^{1/2}}$ distinct \hat{y} for which there exists $(z_w)_{w \in \{0,1\}^t}$ such that $(x, \hat{y}, (z_w)_{w \in \{0,1\}^t}) \in \hat{R}$, contradicting out analysis above.

4. (a) Given an $n \times n$ matrix A with nonnegative integer entries, we produce a circuit that takes as input a permutation π on the set $\{1, 2, \dots, n\}$, and z_1, z_2, \dots, z_n , where each $z_i \in \{0, 1\}^m$, where m is the least positive integer for which 2^m exceeds the largest entry of A . It is clear that the input to this circuit is at most polynomial in the length of the bitstring that describes A . We view each z_i as specifying an integer in $\{0, 1, 2, \dots, 2^m - 1\}$. The circuit then outputs 1 if $z_1 < A[1, \pi(1)]$ and $z_2 < A[2, \pi(2)]$ and $z_3 < A[3, \pi(3)]$, and \dots and $z_n < A[n, \pi(n)]$. Since this is a polynomial-time computation, and the circuit's input is polynomial in the size of A , the overall circuit is polynomial in the size of A . For each particular π , let's count the number of z_1, z_2, \dots, z_n that cause the C to output 1. We can choose any one of $A[1, \pi(1)]$ values for z_1 , any one of $A[2, \pi(2)]$ values for z_2 , etc... Thus the total number of satisfying assignments of C is exactly

$$\sum_{\pi} \prod_{i=1}^n A[i, \pi(i)]$$

which is exactly $\text{PERM}(A)$. We have produced an instance of $\#SAT$, whose answer is $\text{PERM}(A)$, and $\#SAT$ is in $\#P$; thus computing $\text{PERM}(A)$ is in $\#P$.

- (b) Given an instance $G(V, E)$ of $\#CYCLECOVER$, produce the matrix A_G whose rows and columns are indexed by V , with $A_G[u, v] = 1$ iff $(u, v) \in E$, and 0 otherwise. There is an exact correspondence between cycle covers in G and permutations of V for which $(i, \pi(i)) \in E$ for all i . But $\text{PERM}(A_G)$ counts exactly these permutations (any other permutation has $A_G[i, \pi(i)] = 0$ for some i and so does not contribute to the sum). Thus the map $G \mapsto A_G$ is a parsimonious reduction from $\#CYCLECOVER$ to f , which shows that computing the permanent is $\#P$ -hard, and together with (a), it is $\#P$ -complete.