CSC2401 Assignment 1 Solutions

November 7, 2015

These solutions are for your help in studying – they have not been proofread carefully, and may contain errors!

Conventions. For $x \in \{0,1\}^k$ and $1 \leq i \leq k$, we denote by $x[i]$ the i^{th} bit of x, with the convention that indeces begin at 1. We may refer to $x \in \{0,1\}^k$ as a *bit string*, *bit vector*, or simply *string*, with the different names intended to convey a particular usage. Such an x may be transparently used to refer to a subset of a finite set of cardinality k, or a k-bit binary representation of a natural number with the interpretation being inferred by context.

1

Let $\mathcal{L}$ be an **NP**-complete unary language. Then there exists a polynomial $p(n)$ and poly-time reduction f such that for all $x \in \{0,1\}^*$, $x \in \mathbf{SAT} \iff f(x) \in \mathcal{L}$ and $f(x) \leq p(|x|)$. It suffices to describe a poly-time algorithm for **SAT**. We may assume without loss of generality that $\text{range}(f) \subseteq \{1\}^* \cup \{0\}$, since any $y \in \text{range}(f) \setminus \{1\}^*$ may be mapped to 0 while preserving f being a reduction.

Fix a reasonable encoding method of CNF forumulas over $\{0,1\}^*$ and define $|\phi|$ to be the length of the encoding of CNF ϕ. We may also implictly refer to the encoding of ϕ as ϕ alone when inferred by context.

Let $\phi(\vec{x})$ be a CNF formula over variables $\vec{x} = x_1, \ldots x_k$. For a partial assignment μ to $\vec{x}$, let $\phi \lfloor_\mu$ be the CNF obtained by deleting all clauses containing a literal μ assigns to 1 and deleting each literal that μ assigns to 0; if the resulting CNF would contain an empty clause, then $\phi \lfloor_\mu = \bot$, and if the resulting CNF is empty, then $\phi \lfloor_\mu = \top$. Observe that $|\phi \lfloor_\mu| \leq |\phi|$. Suppose a partial assignment μ gives truth values to $x_1, \ldots x_j$ for $0 \leq j < k$; define the *extensions* μ_0, μ_1 as the partial assignments assigning 0 (resp. 1) to x_{j+1} and assigning all other variables according to μ. Let ϵ be the empty partial assignment, assigning truth values to no variables.

We define the complete binary *self-reduction tree* T_ϕ as follows:

- The root is labeled with ϵ.
- If an interior node is labeled with μ, where μ assigns truth values to $x_1, \ldots x_j$, $0 \leq j < k$, then the left and right children are labeled with μ_0 and μ_1 respectively.

Paths and nodes in T_ϕ are in one-to-one correspondence with partial assignments to $\vec{x}$, with leaf nodes corresponding to total assignments. We identify nodes unambiguously using their labelling assignments. For any node μ, $\phi\restriction_\mu$ is unsatisfiable iff $\phi\restriction_\pi$ is unsatisfiable for all descendants π of μ. Thus ϕ is satisfiable iff there exists a leaf μ such that $\phi\restriction_\mu = 1$. We say that a node μ is unsatisfiable if μ cannot be extended to a satisfying assignment for ϕ.

We now give a poly-time algorithm M for **SAT**. Fix an input CNF ϕ over k variables. M operates by performing a depth-first search of T_ϕ, using the reduction f to prune the search by identifying unsatisfiable nodes. M maintains a set U of strings; initially, $U = \emptyset$. M evaluates $dfs(\epsilon)$ on T_ϕ, defined recursively at node μ as follows:

- Compute $\phi\restriction_\mu$ and $f(\phi\restriction_\mu)$.
- If $\phi\restriction_\mu = \top$, return 1.
- If $f(\phi\restriction_\mu) \in U$ or $\phi\restriction_\mu = \bot$, set $U = U \cup \{f(\phi\restriction_\mu)\}$ and return 0.
- Otherwise, compute $a = dfs(\mu_0) \vee dfs(\mu_1)$. If $a = 1$, return 1; otherwise set $U = U \cup \{\phi\restriction_\mu\}$ and return 0.

M accepts iff $dfs(\epsilon) = 1$. Clearly if $dfs(\mu) = 1$ then $\phi\restriction_\mu$ has a satisfying assignment. Conversely, observe that only no-instances of $\mathcal{L}$ are added to U; thus if $dfs(\mu) = 0$ then $\phi\restriction_\mu$ is unsatisfiable. We now show that $dfs(\epsilon)$ can be computed in polynomial time.

To show that each non-recursive step in $dfs(\mu)$ runs in polynomial time, it suffices to show that inclusion in U can be determined in polynomial time. We first observe the invariant that $U \subseteq \text{range}(f)$, so $|y| \leq p(|\phi|)$ for all $y \in U$ and $|U| \leq p(|\phi|) + 1$ (accounting for the lone non-unary string $0 \in \text{range}(f)$). Thus for any μ, testing $f(\phi\restriction_\mu) = y \in U$ can be done in polynomial time by scanning y for each element of U. Thus it suffices to show that the number of nodes on which dfs is computed is at most polynomial in $|\phi|$.

Say that M *visits* a node if M computes $dfs(\mu_0)$ and $dfs(\mu_1)$. Let T'_ϕ be the tree obtained by pruning all subtrees not visited by M. Let N be the number of distinct paths in T'_ϕ. We claim that $N \leq p(|\phi|)$. Suppose for contradiction that $N > p(|\phi|)$. Then we may choose two distinct leaves π and ρ of T'_ϕ such that $f(\phi\restriction_\pi) = f(\phi\restriction_\rho)$ (if no such π, ρ exist the claim is trivial). We may

assume without loss of generality that ϕ precedes ρ in depth-first order and that both $\phi\restriction_\pi$ and $\phi\restriction_\rho$ are unsatisfiable. But then $f(\phi\restriction_\pi) \in U$ after M visits π, contradicting ρ being in T'_ϕ. Since the depth of T_ϕ is at most k, M visits at most $k \cdot N$ nodes, and hence dfs is computed on at most $2 \cdot k \cdot N \leq 2 \cdot k \cdot p(|\phi|)$ nodes, completing the proof. ■

2

2.1

Let Λ denote the empty clause, and for a CNF φ with variable x, let $\varphi \restriction_{x=a}$ denote φ under the partial assignment setting $x = a$ and not assigning any other variables.

Let ϕ be a 2CNF formula over variables $\{x_1, x_2 \dots x_n\}$. Define L_ϕ as the set of literals appearing in ϕ, and define the *implication graph* G_ϕ as the directed graph over $2n$ vertices $V(G_\phi) = L_\phi$ and edges $E(G_\phi)$ containing (a, b) for all $\{\bar{a}, b\} \in \phi$, $(\bar{a}, a)$ for all $\{a\} \in \phi$, and no other edges. We write $a \rightsquigarrow_\phi b$ if there is a directed path from a to b in G_ϕ. We may omit mention of ϕ when a particular formula is clear from context. We note the following straightforward properties of G_ϕ:

- (*) Any edge $(a, b) \in E(G)$ corresponds to a clause $\{\bar{a}, b\} \equiv a \to b$ in the underlying formula, so by transitivity if $a \rightsquigarrow_\phi b$ then ϕ implies $a \to b$.
- (**) By the symmetry of clauses, if $a \rightsquigarrow_\phi b$ then $\bar{b} \rightsquigarrow_\phi \bar{a}$.

The following claim relates the satifiablity of ϕ to certain cycles in its implication graph.

Claim 1 *For any 2CNF ϕ such that $\Lambda \notin \phi$, ϕ is unsatisfiable iff $a \rightsquigarrow_\phi \bar{a} \rightsquigarrow_\phi a$ for some $a \in V(G_\phi)$.*

Proof. The reverse direction follows immediately by observing that if $a \rightsquigarrow \bar{a} \rightsquigarrow a$, then by (*) and (**), any satisfying assignment to ϕ also satisfies $a \equiv \bar{a}$, contradiction.

For the forward direction, fix 2CNF ϕ not containing Λ. We show the contrapositive by strong induction on the number of variables in ϕ. For the base case, ϕ is the empty conjunction and is trivially satisfiable.

For the inductive step, assume ϕ is over variables $x_1, x_2 \dots x_{n+1}$ and that for all literals a, there is no path $a \rightsquigarrow \bar{a} \rightsquigarrow a$. Then there exists a literal such that there is *no* path $a \rightsquigarrow \bar{a}$. Define partial assignment μ such that $\mu(b) = 1$ if $a \rightsquigarrow b$ and $\mu(b) = 0$ if $b \rightsquigarrow \bar{a}$ with all other variables unassigned. μ is well defined, since if $a \rightsquigarrow b$ and $a \rightsquigarrow \bar{b}$ then $a \rightsquigarrow b \rightsquigarrow \bar{a}$ by (**); Then μ satisfies every clause of ϕ containing a literal reachable from a. Furthermore, μ cannot falsify ϕ - if a singleton clause $\{b\}$ were falsified, then $\bar{b} \rightsquigarrow b$ and so $\bar{b}$ and b are both reachable from a contradicting our previous assertion.

Thus $\phi\restriction_\mu$ contains at least one fewer variable, does not contain Λ, and the implication graph $G_{\phi\restriction_\mu}$ is a subgraph of G_ϕ and hence does not contain a path $c \rightsquigarrow \bar{c} \rightsquigarrow c$ for any literal $c \in L(G_{\phi\restriction_\mu})$. Thus by the inductive hypothesis, $\phi\restriction_\mu$ has a satisfying assignment σ which combined with μ yields a satisfying assignment for ϕ. This completes the proof of Claim 1.

Claim 2 $\mathbf{2SAT} \in \mathbf{NL}$.

Proof. We show that $\mathbf{2SAT}^C \in \mathbf{NL}$ by giving a non-determininstic log-space algorithm M for deciding $\mathbf{2SAT}^C$. As indicated by Claim 1, we essentially use the **NL**-algorithm for **PATH**, implicitly using ϕ as the representation of G_ϕ. The full algorithm is as follows:

Let ϕ be the input. Scan the input tape and accept outright if ϕ does not encode a 2CNF formula or contains Λ, and reject if ϕ contains no clauses. Guess a literal x and carry out the **PATH** algorithm for endpoints (literals) $(x, \bar{x})$, rejecting if no path is found in $2n$ steps. Repeat with $(\bar{x}, x)$, accepting if the algorithm succeeds within $2n$ steps.

Clearly the space requirements are no greater than **PATH**. If ϕ is satisfiable then by Claim 1 there is no literal x where $x \rightsquigarrow \bar{x} \rightsquigarrow x$. Then no choice of x in the algorithm can pass both invocations of the **PATH** algorithm. The converse follows similarly from Claim 1.

Since $\mathbf{NL} = \mathbf{CoNL}$ it immediately follows that $\mathbf{2SAT} \in \mathbf{NL}$. ■

2.2

We show that the NL-hard problem $\mathbf{PATH}^C$ is log-space reducible to 2SAT by exhibiting a log-space algorithm M computing a reduction f. Let $\langle G, s, t\rangle$ be an input to PATH with $G = (V, E)$ and $s, t \in V$ (define $f(x)$ as some standard tautology for any malformed input x). M forms the 2CNF ϕ consisting of clauses $\{\bar{a}, b\}$ for each $(a, b) \in E$, and clauses $\{s\}$, $\{\bar{t}\}$. Clearly ϕ is computable in logarithmic space. Observe that if $(a, b) \in E$ then any satisfying assignment to ϕ satisfies $a \to b$. We claim that t is unreachable from s iff ϕ has a satisfying assignment.

For the backward direction, assume that there exists a path $s \rightsquigarrow t$. Then ϕ has a clause $\{\bar{a}, b\}$ for each edge on the path, so any satisfying assignment to ϕ must satisfy $s \to t$ by transitivity. But any such assignment must also satisfy clauses $\{s\}$ and $\{\bar{t}\}$, a contradiction.

For the forward direction, suppose t is not reachable from s. Then there exists a partition S, T of V with $s \in S$ and $t \in T$ and no edge from S to T. Let μ assign all $u \in S$ to 1, and all $v \in T$ to 0. We claim that μ satisfies ϕ. By definition of ϕ we may write $\phi = \phi_S \wedge \phi_T$ where ϕ_S contains $\{s\}$ and $\{\bar{a}, b\}$ for all $a \in S$, and ϕ_T contains all other clauses. Then μ satisfies $\{s\}$ and all $\{\bar{a}, b\} \in \phi_S$ since $b \notin T$ by definition of S, T. Conversely, μ satisfies $\{\bar{t}\}$ and any $\{\bar{a}, b\} \in \phi_T$ since $a \notin S$ by definition of ϕ_T. ■

3

We follow the hint given in the handout, beginning with the following lemma giving an upper bound on the VC-dimension of a collection.

Lemma 3 *Fix finite U and m and let $S = \{S_i\}_m$ where $S_i \in U$ for all $0 < i \leq m$. Then $VC(S) \leq \lfloor \log m \rfloor$.*

Proof. Fix m and $S = \{S_i\}_m$ and suppose to the contrary that $VC(S) = c > \lfloor \log m \rfloor$. Then there exists $X \subseteq U$ of size c which is shattered by S. There are 2^c subsets of X but the set $\{S_i \cap X\}_m$ has at most m distinct elements, contradiction.

We give an alternate definition of VC under the bit-string representation of sets. Fix finite U with cardinality n, For n-bit strings S and X, define $S\lceil X$ (read as S *restricted to* X) as the subsequence of S selected according to the ones of X. For a collection $S = \{S_i\}_m$, define $S\lceil X = \{S_i\lceil X\}_m$. Observe then that X is shattered by collection S iff $S\lceil X = \{0,1\}^k$, where k is the cardinality of X (note that we transparently use the bit-string representation of S as needed).

3.1

Claim 4 *VCdim is in* **NP**.

Proof. Fix finite U of size n and input $\langle S = \{S_i\}_m, k\rangle$ of length $\mathbf{O}(nm)$. Consider an advice string (X, I) where X is an n-bit string and I a sequence of at most $k' = \lfloor \log m \rfloor$ binary integers in $[m]$; clearly the length of such a string is polynomial in nm. Suppose $I = i_1 \dots i_k$, and interpret X as a subset of U. Let R be the predicate $\forall j < k \cdot S_{i_j}\lceil X = j - 1$, where $S_{i_j}\lceil X$ is intepreted as a k'-bit binary integer. It is clear by definition of R that $\langle S, k\rangle$ has a witness iff $\langle S, k\rangle$ in $VCdim$; namely, a set X of size $\geq k$ shattered by S, and an appropriate choice of S_i for each subset of X. That R runs in time polynomial in mn follows from Lemma 3, which gives a $\mathbf{O}(mn)$ upper bound on the number of S_i's to check . ■

3.2

Claim 5 *There is a TM deciding VCdim in time $n^{\mathbf{O}(\log n)}$.*

Proof. The following algorithm satisfies the claim: fix input (S, k) where $S = \{S_i\}_m$, $|S_i| = n$ and $k \leq n$. Compute $k' = \lfloor \log m \rfloor$. If $k > k'$, reject. The algorithm will maintain an array A of k bits, initially all 0. For each $j \in [k]$, perform the following steps: For each index $i \leq n$, scan over bit position i in each element of S and count the number of 1's. If some i has $m/2^j$ 1's, set $A[j] = 1$. Accept if $\bigvee_{j=1}^{k} A[j] = 1$.

It is easy to see that $\bigvee_{j=1}^{k} A[j] = 1$ iff $S\lceil X = \{0,1\}^k$ for some X (namely, $X = \{j_1, j_2 \ldots j_k\}$ where for each $i = 1 \ldots k$, $A[i]$ was set at iteration j_i). The time bound follows from Lemma 3, which gives us a $\mathbf{O}(\log nm)$ bound on the number of iterations. ■

We now address the conjecture that VCdim is not **NP** complete. Were this the case, by Claim 5 we would have that **NP** is contained in $\mathbf{DTIME}[\mathbf{n}^{\mathbf{O}(\log \mathbf{n})}]$. While not refuting the $\mathbf{P} \neq \mathbf{NP}$ conjecture in and of itself, it is widely held that there are hard problems in **NP** that do not admit 'near'-polynomial time algorithms.

4

4.1

Given the definitions of VC-dimension and Σ_3^p, it will suffice to show that there is a poly-time predicate $R(\langle C, k\rangle, X, X', i)$ computing (informally) $\forall x \in X \cdot C(x, i) = 1 \iff x \in X'$, where $X' \subseteq X \subset U$, $|X| = k$, $i \leq m$ and C a circuit as given in the problem.

In order to properly specify R, the representations of its inputs must be carefully specified to be polynomial in $|C| \in \mathbf{O}(\log^c mn)$. Fix $k' = \lfloor \log m \rfloor$. Recall from Lemma 3 that k' is an upper bound on $VC(S)$ for a collection of size m.

We represent X as a sequence $x_1, x_2 \ldots x_k$ of binary numbers representing elements of U (padding with 0's up to length $\mathbf{O}(k' \log n)$). We represent $X' \subseteq X$ as a k'-bit vector where $X'(j) \iff x_j \in X'$. Clearly then R is computable in time polynomial in N, by checking for each bit j $(0 < j \leq k')$ of X' that $C(x_j, i) = X'(j)$. An initial step rejects if $k > k'$ or X has cardinality $< k$.

It follows from the definition of R and the lengths bounds given for its inputs that, for any instance $\langle C, k\rangle$,

$$\langle C, k\rangle \in VCdimSuccinct \iff \exists X \forall X' \exists i \cdot R(\langle C, k\rangle, X, X', i)$$

hence $VCdimSuccint \in \Sigma_3^p$. ■

4.2

We show that $\Sigma_3\mathbf{SAT} \leq_\mathrm{p} VCdimSuccinct$.

We define a reduction f from $\Sigma_3\mathbf{SAT}$ to $VCdimSuccinct$ as follows. Say the formula is $\exists x \forall y \exists z \phi(x, y, z)$, and assume without loss of generality that the $\forall y$ is only quantified over strings y of positive Hamming weight (a simple trick achieves this). Assume x, y, z all have length n.

The circuit will encode a matrix or table, where the rows correspond to the universe elements, and the i^{th} column is the indicator for the i^{th} set. There are $n2^n$ rows, with 2^n groups corresponding to all x's, and each group having n rows corresponding to the coordinates of y. There is a column for each triple (x, y, z).

The matrix is block diagonal: it is 0 except when the row group label x matches the x from the column triple. Within the block corresponding to a particular x, you have length-n columns corresponding to each possible (y, z). If the formula accepts (x, y, z) (that is, $\phi(x, y, z)$ is true), then write y in that column, otherwise write the all-0 string of length n. It is straightforward to

check that there is a small circuit $C(a, b)$ that outputs the (a, b) entry of the matrix.

We want to show that the set system defined by C has VC dimension n if and only if the formula is a yes-instance. For the easy direction, assume that the formula is a yes-instance, and suppose that w is the value of x such that $\forall y \exists z \phi(w, y, z)$ is true. Look at the block corresponding to the value of $x = w$. Since this is a yes-instance, for every value of y there is at least one z-value such that $\phi(w, y, z)$ is true, and thus by the definition of the matrix, there will exist 2^n columns within this block that contain all possible values of y. Thus there is a shattered set of size n.

For the hard direction, suppose that there is a shattered set of size n. We want to show that the formula is a yes-instance. To see this, we show first that the shattered set must be exactly the full group corresponding to some x. Then you need the assumption that we're only quantifying over nonzero y's, because the $y = 0^n$ pattern will show up even when the formula doesn't evaluate to true.

5

This problem is to prove that $SPACE(n) \neq NP$.

We will prove that there is a language in NP that is not in $SPACE(n)$ since NP is closed under polynomial time reductions and $SPACE(n)$ is not. Let L be any language over $\{0,1\}$. We define a new language $Pad(L, f(n))$ as follows:

$$Pad(L, f(n)) = \{x\#1^{f(|x|)-|x|} \mid x \in L\}$$

We first claim that if $Pad(L, f(n))$ is in NP, then $L \in NP$ for all $f(n) \in O(p(n))$ where p is some polynomial.

To prove this, suppose that $Pad(L, f(n)) \in NP$. Then we can decide $Pad(L, f(n))$ in nondeterministic polynomial time. We can then decide L in nondeterministic polynoimal time as follows. On input x, append $\#1^{f(|x|)-|x|}$ to x and decide whether $x\#1^{f(|x|-|x|} \in Pad(L, f(n))$. If so, accept x. If not, reject x. Since $f(n) \in O(p(n))$ for some polynomial p, $Pad(L, f(n))$ is computable in nondeterministic polynomial time and thus $L \in NP$.

We will now use this claim to show that NP and $SPACE(n)$ are different. Assume for contradiction that $NP = SPACE(n)$. Let L be a language in $SPACE(n^2)$ but not in $SPACE(n)$. We know that such a language exists by the space hierarchy theorem. By our assumption, L is not in NP, and therefore by our lemma above, $Pad(L, n^2)$ is not in NP as well. Let M_L be a TM which decides L using $O(n^2)$ space and consider the following TM, M'. M' on input x proceeds as follows. If $x \neq y\#1^{|y|^2-|y|}$, then reject. Otherwise run M_L on y; accept if and only if M_L accepts.

Clearly M' decides $Pad(L, n^2)$ in linear space since M_L requires at most $O(|y|^2)$ space and $|x| \in O(|y|^2)$. Therefore, $Pad(L, n^2) \in SPACE(n)$. Therefore, $Pad(L, n^2) \in NP$, and we have reached a contradiction.