Revisions

Exercise 1  Ladner’s theorem

We are going to show the following theorem:

Theorem (Ladner, 1975). If $P \neq NP$, then there exists a language $L \in NP \setminus P$ that is not NP-complete.

For this, we define a language $SAT_H$ depending on $H : \mathbb{N} \to \mathbb{N}$, for padding the input.

Definition. $SAT_H := \{ \phi 01^n_H(n) ; \phi \in SAT \land n = |\phi| \}$

Now, let us define a particular function which will imply the result using a diagonalization technique:

Definition. $H(n)$ is the smallest number $i < \log \log n$ (for $n$ large enough) such that for every $x \in \{0,1\}^*$ with $|x| < \log n$, the $i$-th machine $M_i$ outputs $SAT_H(x) \in \{0,1\}$ within $i|x|^i$ steps. If it does not exist, then $\log \log n$ is taken instead.

1. Show that this function is well-defined,
2. Show that it can be computed in polynomial time,
3. How many representations does a Turing machine have?
4. If $SAT_H \in P$, show that $H(n)$ is bounded.
5. Show that the reciprocal is true.
   Hint: recall the pigeon-hole principle.
6. If $SAT_H \in P$, show that $SAT \in P$.
7. If it is NP-complete, let $f$ be a reduction from $SAT$ to $SAT_H$. Show that for a formula $\phi$ of large size $n = |\phi|$, if $\psi,m$ are such that $f(\phi) = \psi 01^n m_H(m)$ with $m = |\psi|$, then $m = O(n^{1/k})$ for any $k > 0$.
8. Give a polynomial time algorithm for SAT in this case.

Exercise 2  P-completeness

A language is in NC when it can be decided by poly size and polylog $(\log^k(n))$ depth circuits. It means intuitively that it is efficiently solvable using parallel algorithms. The main conjecture regarding this class, is that not all P problems have efficient parallel algorithms, i.e. $NC \subsetneq P$.

Definition. A language $L$ is P-complete if for all $L' \in P$, there exists a reduction $f$ from $L$ to $L'$, with polynomially bounded output, and such that the sets $\{(x,i) ; i \leq |f(x)|\}$ and $\{(x,i) ; f(x)_i = 1\}$ are decidable in log space.

1. Justify the inclusion $NC \subseteq P$.
2. Show that if there is a P-complete language in NC, then $P = NC$. Show the same for $L$ instead of NC.
3. Show that the following language ("circuit evaluation") is P-complete:

\[ \{(C,x) ; C \text{ has } n \text{ inputs and single output, } C(x) = 1\}. \]
Exercise 3  A natural complete problem... with circuits!

If \( S = \{S_i\} \) is a finite collection of concepts (=subsets) of a finite universe \( U \), the Vapnik-Chervonenkis (VC) dimension, denoted \( VC(S) \) is the size of the largest \( X \subseteq S \) which has the following property:

\[
\forall X' \subseteq X, \quad \exists i, \quad X' = X \cap S_i.
\]

We are interested in studying this dimension — which is a major indicator of difficulty in statistical learning — in cases where the concepts may have exponential sizes, with regard to their description. Typically, this is the case for monomials in \( \{0,1\}^n \) for instance, where one bit encodes one concept.

A boolean circuit \( C \) succinctly represents \( S \) if \( S_i = \{x; \; C(i,x) = 1\} \) for all \( i \). Finally, the problem VC-dimension is the set of \( (C,k) \) where \( C \) represents a collection \( S \) s.t. \( VC(S) \geq k \).

1. Show that if \( (C,k) \) is in VC-dimension, then \( k \leq |C| \).
2. Show that this problem is in \( \Sigma_p^3 \).
3. How would you start a proof that it is \( \Sigma_p^3 \)-complete?