# CS41 Lab 11: Polynomial-Time Verifiers and Polynomial-Time Reductions

November 17, 2022

This week, we've started to understand what makes some problems seemingly hard to compute. In this lab, we'll consider an easier problem of *verifying* that an algorithm's answer is correct. Recall that a *decision problem* is a problem that requires a YES or NO answer. Alternatively, we can describe decision problem as a set $L \subseteq \{0,1\}^*$; think of $L$ as the set of all YES inputs i.e., the set of inputs $x$ such that one should output YES on input $x$. Let $|x|$ denote the length of $x$, in bits.

**Polynomial-time Verifiers**. Call $V$ an efficient *verifier* for a decision problem $L$ if

1. $V$ is a polynomial-time algorithm that takes two inputs: $x$, and $w$.
2. There is a polynomial function $p$ such that for all strings $x$, $x \in L$ if and only if there exists $w$ such that $|w| \leq p(|x|)$ and $V(x, w) =$ YES.

$w$ is usually called the *witness* or *certificate*. Think of $w$ as some *proof* that $x \in L$. For $V$ to be a polynomial-time verifier, $w$ must have size some polynomial of the input $x$. For example, if $x$ represents a graph with $n$ vertices and $m$ edges, the length of $w$ could be $n^2$ or $m^3$ or $(n+m)^{100}$ but not $2^n$.

Consider this lab a **success** if you complete problems 1-3 and make progress on problems 4 and 5. Do not feel the need to formally write up solutions.

1. **Polynomial-time reductions.** In class yesterday, we saw the following lemma:

   **Lemma.** *For any graph $G = (V, E)$, $S \subseteq V$ is a vertex cover if and only if $V \setminus S$ is an independent set.*

   Give the following polynomial-time reductions:

   (a) VERTEX-COVER $\leq_P$ INDEPENDENT-SET.

   (b) INDEPENDENT-SET $\leq_P$ VERTEX-COVER.

2. **Transitivity of polynomial-time reductions.** Show the following:

   $$\text{If } A \leq_P B \text{ and } B \leq_P C \text{ then } A \leq_P C.$$

3. **Verifier Debugging.** Consider the THREE-COLORING problem: Given $G = (V, E)$ return YES iff the vertices in $G$ can be colored using at most three colors such that each edge $(u, v) \in E$ is *bichromatic*.

   Consider the following verifier for THREE-COLORING. The witness we request is a valid three coloring of the undirected graph $G = (V, E)$, which is specified as a list of two-digit binary strings $w = w_1 w_2 \dots w_k$ where we interpret

   $$w_i = \begin{cases} 00, & \text{vertex } i \text{ is colored BLUE} \\ 01, & \text{vertex } i \text{ is colored GREEN} \\ 10, & \text{vertex } i \text{ is colored RED} \end{cases}$$

```
threeColoringVerifier(G = (V, E), w)
for each w_i in w
    if w_i = 11
        return NO
    for j from i + 1 to len(w)
        if w_i = w_j and (i, j) ∈ E
            return NO
return YES
```

This verifier is not quite right.

Give an example witness $w$ and graph $G$ which is *not* three-colorable, such that

$$\text{threeColoringVerifier}(G, w) = \text{YES}$$

4. Rewrite threeColoringVerifier so that it is a valid verifier for Three-Coloring.

5. Give polynomial-time verifiers for the following problems, none of which are known to have polynomial-time algorithms.

   (a) Independent-Set.

   (b) Vertex-Cover.

   (c) Sat.

   (d) Factoring. Given numbers $n, k$ written in binary, output YES iff $n$ is divisible by $d$ for some $1 < d \leq k$.

   (e) Not-Factoring. Given numbers $n, k$ written in binary, output YES iff $n$ is **NOT** divisible by $d$ for any $1 < d \leq k$.

   **Hint:** The following problem **is solvable** in polynomial time.[1]

   Primes: Given a number $n$ written in binary, output YES iff $n$ is a prime number.

6. Prove the Lemma from problem 1.

[1] This actually wasn't known until 2002, when Agrawal, Kayal, and Saxena created the AKS primality test. Kayal and Saxena were undergraduates at IIT Kanpur at the time; Agrawal was their advisor.