# ZERO KNOWLEDGE PROOFS

GLENN SUN

OLGA RADKO MATH CIRCLE ADVANCED 2

April 24, 2022

**Problem 1.** Suppose you are color-blind and cannot differentiate red and green. You are skeptical that they are actually different, and your friend who is not color-blind wants to convince you that they are different.

For this activity, we have prepared some index cards that are blank on one side, and have "red" or "green" written on the other. With a partner, choose two cards and hold them so that only one of you can see the colors.

1. If you can see the colors, your job is to convince the color-blind person that you can distinguish the colors, regardless if they are actually different.
2. If you are color-blind, your job is to correctly agree that the colors are different or correctly detect that you are being lied to.

Can the color-blind person successfully do their job? You can ask each other questions, use randomness, or do whatever else you want in the protocol.

You and your partner just gave an *interactive proof* that two objects are different.

**Definition 1.** An interactive proof for a problem $f : X \to \{\mathsf{T}, \mathsf{F}\}$ is a protocol between an unbounded-time *prover* and a polynomial-time *verifier* such that

1. For all $\mathsf{T}$-instances, there exists something the prover can say to make the verifier agree that $f(x) = \mathsf{T}$.
2. For all $\mathsf{F}$-instances, the verifier finds a mistake in the prover's argument with at least 99% probability. (The probability is not dependent on the instance.)

**Problem 2.**

1. In your protocol, who was the prover and who was the verifer? What is $X$?
2. Given two differently colored cards, can the prover always make the verifer agree?
3. Given two identical cards, what is the probability that the verifier detects the lie? How many times should you repeat the protocol to get a 99% chance of deduction?

**Definition 2.** The set of problems for which interactive proofs exist is called $\mathsf{IP}$.

**Problem 3.** Recall the definitions of $\mathsf{P}$ (solvable in polynomial time) and $\mathsf{NP}$ (proofs of $\mathsf{T}$-instances checkable in polynomial time).

1. Show that $\mathsf{P} \subset \mathsf{IP}$.
2. Show that $\mathsf{NP} \subset \mathsf{IP}$.

Notice that in the above problem, you didn't need the fact that randomness is allowed in $\mathsf{IP}$. In Problem 9, you will show that interactive proofs without randomness are exactly as powerful as $\mathsf{NP}$. Does randomness give interactive proofs more power?

Most computer scientists believe the answer is yes. The graph nonisomorphism problem is a problem that most computer scientists do not believe is in $\mathsf{NP}$, but we will soon give an interactive protocol for it. Two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$ are isomorphic (Greek for *same form*) if there exists a bijection $\varphi : V_G \to V_H$ between the vertices such that $\{u, v\}$ is an edge in $G$ if and only if $\{\varphi(u), \varphi(v)\}$ is an edge in $H$. The problem asks: given $G$ and $H$, are they *not* isomorphic?

**Problem 4.**

1. Draw two graphs that are isomorphic, and two graphs that are not isomorphic.
2. Show that the graph isomorphism problem, which asks if two graphs *are* isomorphic, is in $\mathsf{NP}$.
3. Could a similar proof work to show that two graphs are *not* isomorphic? Can you see why most computer scientists believe graph nonisomorphism is not in $\mathsf{NP}$?

Consider protocol to solve graph nonisomorphism is as follows:

1. The verifier picks $G$ or $H$, randomly changes the vertex labels (and updates the edges to be consistent), and sends the graph over.
2. The prover, who wants to prove that $G$ and $H$ are not isomorphic, says if the verifier sent a version of $G$ or a version of $H$. (Check understanding: How?)
3. The verifier agrees if the prover guessed correctly, and disagrees otherwise.

**Problem 5.** Show that this is a valid interactive proof by doing the following:

1. Check that on $\mathsf{T}$-instances, the prover can always make the verifier agree.
2. Compute the probability that the verifier disagrees on $\mathsf{F}$-instances. How many times should this protocol be run to reach 99% probability of lie detection?

Next, we introduce the concept of zero-knowledge. Intuitively, a proof is zero-knowledge if the verifier doesn't gain any knowledge about why anything is true. You can imagine many reasons why this is useful, for example, proving that you know a password without telling the verifier what it is. Formally, we make the following definition.

**Definition 3.** An interactive proof is called zero-knowledge if for all $\mathsf{T}$-instances, the verifier can simulate the prover's responses by themself. The set of problems with a zero-knowledge proof is $\mathsf{ZKP}$.

**Problem 6.**

1. Show that the color-guessing protocol is zero-knowledge.
2. Show that if the protocol you gave to prove $\mathsf{NP} \subset \mathsf{IP}$ is zero-knowledge, then $\mathsf{P} = \mathsf{NP}$. (Hence, we believe that protocol to be not zero knowledge.)
3. Is the protocol for graph nonisomorphism zero-knowledge?

Despite the fact that the above interactive protocol for $\mathsf{NP}$ problems isn't zero-knowledge, it is still true that $\mathsf{NP} \subset \mathsf{ZKP}$. We will prove this now by giving a $\mathsf{ZKP}$ protocol for an $\mathsf{NP}$-complete problem. Recall that last week, we noted that graph 3-coloring is $\mathsf{NP}$-complete. Consider the following protocol using the provided printouts:

First, the prover draws a 3-coloring on a piece of paper, as well as all 6 permutations of the 3 colors. Then, we repeat the following protocol until the verifier is satisfied.

1. The prover randomly picks one of the six drawings and covers up each vertex with a little piece of paper.
2. The verifier points to an edge that they want revealed.
3. The prover reveals the colors of the endpoints of the edge.

**Problem 7.** In this problem, we show that $\mathsf{NP} \subset \mathsf{ZKP}$ by verifying that the above is a valid $\mathsf{ZKP}$ protocol for 3-coloring.

1. Use the printouts to run this protocol a few times with a partner.
2. Show that the prover can always make the verifier agree on $\mathsf{T}$-instances.
3. What is the probability of detecting a lie in one round of the protocol? How many times do we need to run the protocol in order for a 99% chance of lie detection? (Hint: $1 - x \leq e^{-x}$ is a good approximation for small $x$.)
4. Show that the protocol is zero-knowledge by saying how the verifier can simulate the prover's responses.

In fact, it has been shown that (under standard cryptographic assumptions) $\mathsf{ZKP} = \mathsf{IP}$! But this proof is a bit beyond the scope of this worksheet. $\mathsf{NP}$ is already an extremely impactful class of problems to have zero-knowledge proofs for, and is sufficient for many applications.

**Problem 8.** Give a zero-knowledge proof for the following problems:

1. Graph isomorphism: Opposite of the nonisomorphism problem discussed before. $\mathsf{T}$-instances (where the verifier should always agree) are now pairs of isomorphic graphs.

2. Hamiltonian cycle: Given a graph $G$, is there a way to walk around the graph, using every vertex exactly once?

3. Discrete log: Given $0 \leq g, y < p$ for some prime $p$, is there $x$ such that $g^x \equiv y \pmod{p}$? (You may need Fermat's little theorem: $g^{p-1} \equiv 1 \pmod{p}$ for all $g$.)

**Problem 9.** This problem goes back to explore the motivation for the definition of $\mathsf{IP}$.

1. Let $\mathsf{DIP}$ be the set of problems with deterministic interactive proofs: that is, for all $\mathsf{F}$-instances, the verifier has to find a mistake in the prover's argument 100% of the time. Show that $\mathsf{DIP} = \mathsf{NP}$. This shows why we want randomness in the definition.

2. Show that for any polynomial $p(n)$, the class $\mathsf{IP}$ is unchanged by requiring lie detection with anywhere between $\frac{1}{p(n)}$ and $1 - 2^{-n}$ probability. This explains the seemingly arbitrary choice of 99% success probability.

3. What happens if you only require lie detection with probability $\frac{1}{2^n}$?