Today

- Interaction, Proofs and Knowledge - Cryptographic motivation.
- Interactive proofs and Arthur-Merlin games.
- Classes IP, AM (and MA).

Formalizing knowledge and proofs

- Can we have a secure login protocol?
- What does it take?
- User X proves to server that he/she *is* X.
- But any eavesdropper should not be able to prove that he/she is X, even after tapping the entire conversation.

Concepts

Proof: Need to convince server that you are X.

Knowledge: Should not reveal any knowledge to server/ eavesdropper other than that you are X.

Complexity theory helps define these notions mathematically. Today we'll focus on proofs.

To have any hope of success - the proof needs to be interactive. Modelling a challenge-response scheme.

Interaction

- Interaction between two entities: Prover and Verifier.
- Parties have common information x.
- Parties, in particular, V has access to randomness.
- Modelled by a series of functions: $P^{(1)}, \ldots, P^{(k)}$ and $V^{(1)}, \ldots, V^{(k)}$.
- $q_i = V^{(i)}(x, R, a_1, \ldots, a_{i-1})$, $a_i = P^{(i)}(x, q_1, \ldots, q_i)$,

Interactive computation

V also has a verdict function $\mathrm{Verdict}(x, R, a_1, \ldots, a$

$V = (\mathrm{Verdict}, V^{(1)}, \ldots, V^{(k)})$ recognizes a language L with completeness c and soundness s if for every x:

$x \in L$ implies there exists a prover $P = (P^{(1)}, \ldots, P^{(k)})$ such that V accepts with probability at least c.

$x \notin L$ implies for every prover P, V accepts with probability at most s.

So the computation is trying to prove membership in L. Such a proof is an "interactive proof".

Goal: Study what kind of L have interactive proofs if verifier is only allowed probabilistic

polynomial time computations. (Prover not computationally bounded.)

Simple example of interactive proof

- Can interactively prove to a color-blind person that notion of color exists (two different pieces of paper that are otherwise identical are distinguishable due to color).
- Take two pieces of paper of different color.
- Prover claims one is Red and other Green.
- Verifier takes Red paper into left hand and Green one into right hand; hides both hands behind its back; and randomly decides to swap them or not. V knows if it swapped or not, but Prover has only a 50% chance of guessing correctly.

- V now reveals the papers in the two hands. If the prover can tell which one is Red and which one Green, then the papers must have been distinguishable (with some confidence).

Central ingredients

- Interaction: No written proof would have worked.
- Randomness: If V was deterministic and P knew its algorithm, then P could cheat.
- Secrecy: V should be able to hide information from P to achieve soundness? Surprisingly, not needed, in mathematical scenarios!

Graph non-isomorphism

GNI $= \{\langle G, H\rangle | \forall \pi \in S_n, G \neq \pi(H)\}$.

- Easy to prove G is isomorphic to H.
- How to show non-isomorphism?
- Can do interactively - mimicking color-non-isomorphism proof.

Interactive proof for GNI

- V picks random permutation $\pi \in S_n$ and a random $F \in \{G, H\}$.
- V sends $\pi(F)$ to P and asks P to guess if $F = G$ or $F = H$.
- If P can guess correctly, then V accepts.

Completeness $= 1$; Soundness $= 1/2$.

Basic attributes

- # of rounds k: Two, Constant, Polynomial.
- Error: One-sided vs. Two-sided.
- Randomness: Public vs. Private. (Secrecy is important?)

Surprising early results

- Two-rounds = Constant rounds.
- One-sided = Two-sided. (while preserving # rounds.)
- Public coins = Private coins (preserving # rounds.)

Classes

- $\langle[\rangle k]$: Class of languages with $k(n)$ round verifiers with completeness $2/3$ and soundness $1/3$.
- IP = IP[poly].
- $[k]$: Class of languages with $k(n)$ round verifiers, where $V^{(i)}(\cdots) = R_i$, where $R = R_1 \circ R_2 \cdots R_k$. -1
- A = Arthur, M = Merlin. Arthur only tosses coins to challenge Merlin, but is not holding any secrets!
- Refined notation: E.g., AMAMA =

Languages involving five "half-rounds" with A going first and last.

- AM = 1-round with Arthur going first.
- MA = 1-round with Merlin going first.

History

IP defined based on cryptographic motivations by Goldwasser, Micali, and Rackoff (ca. 1982-1984). Profound paper, both for definition of IP, and of Zero-knowledge (something we may get to later).

AM defined by Babai. (Somewhat later, but independently.)

In retrospect, what matters most is number of rounds. AM has become synonymous with 2-rounds, IP with poly rounds.

How do they relate to other complexity classes

AM - fairly well-understood.

- AM $= \mathrm{BP} \cdot \exists \cdot \mathrm{P}$.
- AM $\subseteq \mathrm{NP}/_{\mathrm{poly}}$.
- AM $\subseteq \Pi_2^P$.
- co-NP $\subseteq AM$ implies PH collapses.
- AM $=$ co-AM implies PH collapses.
- AM $=$ IP? (quite open till 1990).
- IP $=$ co-IP? (ditto).

Today

Most AM properties provable given what we know.

- AMAM $=$ AM (Exchange of quantifiers - similar to that in Toda's theorem.)
- co-NP in NP/poly implies hierarchy collapses. Similar to Karp-Lipton.

We'll focus on public vs. private coins.

The optimal prover

- Given a fixed verifier, what should a prover do?
- Can figure out what to do, optimally, by computing the following quantity:
- Given a history of interactions so far, what is the highest probability, over all provers, of the verifier accepting.
- Can compute this by induction on number of remaining rounds.
- Prover that does this is the optimal prover.

IP $\subseteq$ PSPACE

Simple consequence of the explicit form of the optimal prover:

Proposition: IP $\subseteq$ PSPACE.

Proof: Can compute "probability of acceptance by optimal responses" in PSPACE.

Lets draw the probability tree:

Root node computes probability of V's acceptance p.

Left Child p_0: Prob. acceptance given first coin is 0.

Right Child p_1: Prob. acceptance given first coin is 1.

p_{00}: Prob. acceptance given first two coins are 0. etc.