Tik-79.511 Special Course on Cryptology / Zero Knowledge:

Interactive Proof Systems

Petteri Kaski

petteri.kaski@hut.fi

Proofs and proof systems

- ▷ A proof can be viewed as an *interaction* between two parties
  - Peggy the *prover*
  - Victor the *verifier*
- ▷ Peggy wants to convince Victor of the validity of some mathematical claim.
- ▷ Victor wants to be convinced, but usually not on false grounds.

Proofs and proof systems

- ▷ The Solovay–Strassen primality test is a probabilistic primality test that detects compositeness with probability $1/2$.
- ▷ Suppose you run 1000 independent Solovay–Strassen tests for p, which all fail to detect compositeness.
- ▷ Is this a "proof" that p is prime?

Proofs and proof systems

▷ A *proof system* is a framework that Peggy uses in her attempt to convince Victor.
Examples:

- write the proof up and send the manuscript to Victor
- Peggy has a black box that Victor trusts to run the Solovay–Strassen test

▷ Completeness:
Peggy can always convince Victor of a truly valid assertion.

▷ Soundness:
Peggy can never convince Victor of an invalid assertion.

The language recognition problem

- ▷ Let Σ be a finite alphabet of symbols, and denote by Σ^* the set of all finite strings over Σ.
- ▷ A *language* over Σ is a subset $L \subseteq \Sigma^*$.

$$L = \{x \in \{0,1\}^* \ : \ x \text{ is a composite number}\}.$$

- ▷ Suppose $x \in \{0,1\}^*$ is known both to Peggy and Victor.
- ▷ Peggy wants to convince Victor that $x \in L$.
- ▷ Peggy and Victor are modelled using Turing machines.

Applications of language recognition

- Computational complexity:
  - How "hard" it is to recognize a language? **P**, **NP**, ...
  - Vary the resources allocated for Peggy and Victor (e.g. Victor is polynomial time in the input size).
  - Vary the level of interaction allowed between Peggy and Victor.
- Cryptography:
  - Victor is unable to decide language membership by himself, yet Peggy wants to convince him.
  - How much information (apart from $x \in L$) must Peggy give to Victor to convince him?

Interactive proof systems

[Goldwasser, Micali, Rackoff 85]; independently [Babai85].

Victor and Peggy share a common input $x \in \Sigma^*$.

- ▷ Victor is a polynomial time deterministic Turing machine V.
- ▷ Victor has access to a random source (e.g. a fair independent coin).
- ▷ Peggy is an unrestricted Turing machine P.
- ▷ Victor and Peggy exchange messages for a polynomial number of rounds.

In the end, Victor has to decide whether he believes $x \in L$.

Interactive proof systems

Let $L \subseteq \Sigma^*$ be a language.

Definition. A pair (V, P), standardized as before, is an *interactive proof system* for L with *error probability* ϵ if,

1. (Completeness)
   For every $x \in L$, V accepts with probability $\geq 1 - \epsilon$; and
2. (Soundness)
   For every $x \notin L$, and *every Turing machine* P' *replacing* P,
   V accepts with probability $\leq \epsilon$,

where the probabilities are calculated over V's coin tosses.

The class $\mathbf{IP}$

Definition. A language $L \subseteq \Sigma^*$ is in the class $\mathbf{IP}$ if it has an interactive proof system with error probability $1/3$.

If the number of rounds required is bounded by $t(n)$ for inputs of length n, then we say that L is in $\mathbf{IP}(2t(n))$.

Example. The language

$$L = \{x \in \{0,1\}^* \ : \ x \text{ is a composite number}\}$$

is in $\mathbf{IP}(2)$:

Peggy sends to Victor the factorization $x = p \cdot q$ of a composite x, which Victor can verify by multiplication (in polynomial time).

Graph nonisomorphism

Input: Graphs $G_1 = (V, E_1)$, $G_2 = (V, E_2)$.

Repeat for n rounds:

1. Victor flips a coin to select either $i = 1$ or $i = 2$.
2. Victor generates a random permutation $\pi \in \mathrm{Sym}(V)$.
3. Victor sends the graph $G' = \pi G_i \stackrel{\mathrm{def}}{=} (V, \pi E_i)$ to Peggy.
4. Peggy computes $G' = \pi G_j$ and sends j to Victor.
5. Victor rejects immediately if $i \neq j$.

If all rounds succeed, Victor accepts.

Parallel amplification

- ▷ The error probability $\epsilon \leq 1/3$ of an interactive proof system can be made arbitrarily small.
- ▷ For input $x \in \Sigma^*$, perform $q(|x|)$ runs, q polynomial, of the original protocol in parallel (each protocol run is independent).
- ▷ Decide according to the majority outcome.

$\Rightarrow$ The error probability becomes exponentially small in the input size, e.g., $\epsilon = 2^{-|x|}$.

Simulation in polynomial space

- ▷ Suppose (V, P) is an interactive proof system for $L \subseteq \Sigma^*$ with error probability ϵ.
- ▷ P is unrestricted.
- ▷ However, the behaviour of V during the protocol can be simulated in polynomial space
  - for every reply that V receives from P; and
  - for every coin flip sequence.
- ⇒ A polynomial space Turing machine can determine the reply that causes the maximum acceptance probability for V.
- ⇒ P can be assumed to be polynomial space.
- ⇒ $\mathbf{IP} \subseteq \mathbf{PSPACE}$.

Arthur–Merlin proof systems

Arthur–Merlin proof systems are a family of interactive proof systems in which

1. on each round, the verifier (King Arthur) flips his coin a number of times and sends the coin flip results to the prover (Merlin the Wizard), who produces a reply.
2. in the end, based on Merlin's replies, Arthur decides either to accept or reject.

$\Rightarrow$ An Arthur–Merlin proof system is an interactive proof system with a public coin.

The classes $\mathbf{AM}$ and $\mathbf{AM}(\text{poly})$

- A language is in the class $\mathbf{AM}(\text{poly})$ if it has an Arthur–Merlin proof system with error probability 1/3.
- A language is in the class $\mathbf{AM}(2t(n))$ if the number of rounds in the protocol is bounded by $t(n)$ for inputs of length n.
- $\mathbf{AM} \stackrel{\text{def}}{=} \mathbf{AM}(2)$
- $\mathbf{AM}(3) = \mathbf{AMA}$, $\mathbf{AM}(4) = \mathbf{AMAM}$, $\ldots$
- $\mathbf{MA}(3) = \mathbf{MAM}$, $\mathbf{MA}(5) = \mathbf{MAMAM}$, $\ldots$

The $\mathbf{AM}$ hierarchy collapses

- $\mathbf{AM}(2t(n)+2) \subseteq \mathbf{AM}(2t(n))$ for $t(n) \geq 1$
  - Swap initial A-M-A-M sequence to A-A-M-M = A-M and parallelize, decide according to the majority of outcomes.
- Speedup theorem [Babai,Moran88]:

$$\mathbf{AM}(2t(n)) \subseteq \mathbf{AM}(t(n))$$

  for every $t(n) \geq 2$.

- For every $k \geq 1$, $\mathbf{AM} \stackrel{\text{def}}{=} \mathbf{AM}(2) = \mathbf{AM}(2k)$.

Public vs. private coin tosses

- ▷ An interactive proof system for L with private coin tosses and $t(n)$ rounds can be simulated by an Arthur–Merlin proof system with $t(n)+2$ rounds. [Goldwasser, Sipser 86]

$\Rightarrow$ $\mathbf{IP}(2t(n)) \subseteq \mathbf{AM}(2(t(n)+2))$

$\Rightarrow$ $\mathbf{AM}(\mathrm{poly}) = \mathbf{IP}$.

- ▷ $\mathbf{AM}(2(t(n)+2)) = \mathbf{AM}(2t(n))$ by the speedup theorem.