Tik-79.511 Special Course on Cryptology / Zero Knowledge: Comments on Kaski: "Rudiments of Interactive Proof Systems"

Johan Wallén

September 27, 2001

1 Introduction

In this document, we give some selected comment on the survey [2] on interactive proof systems. We refer to the survey for definitions, terminology, and notation. Unless states otherwise, all references refers to the corresponding items in [2].

In general, the survey gives a good overview of interactive proof systems. It is generally well written and readable. The introduction (Section 1) and the section on interactive proof systems (Section 2) makes the intuition clear and presents clear formalizations of the concepts. The formalism chosen is much clearer than the one based on multiple-string Turing machines.

The examples in Section 3 are well chosen, since they illustrate the asymmetry in the definition of interactive proofs. In addition, the protocol for Graph Nonisomorphism is a quite canonical example of simple interactive proofs.

The proof that $\mathbf{IP} \subseteq \mathbf{PSPACE}$ could probably be simplified and made more intuitive by considering a suitable computation tree of the simulation and arguing that the problem can be solved by traversing the (polynomially deep) tree in polynomial space. This is, of course, essentially the same as the proof given.

The proof of the Collapse Theorem (Section 4) is fairly intuitive despite its unavoidable technicality.

The proof that $\mathbf{IP}[2t(n)] \subseteq \mathbf{AM}[2t(n)+2]$ is not very clear. What is the intuition behind the proof? Is it not that the **AM** prover can simulate the **IP** protocol and convince the **AM** verifier that the **IP** verifier would have accepted? How is this achieved using one extra round and the **AM** verifier's coin tosses?

One of the motivations behind **AM** was to classify some problems "just above" **NP** (just as **BPP** is "just above" **P**). In addition, there is a simple classification of $\mathbf{IP}(=\mathbf{PSPACE})$ in terms of polynomial-time randomized nondeterministic machines with bounded (two-sided) error probability. These things might be worth mentioning in the survey.

It would probably be better to define the number of rounds differently. This way, the '2's in $\mathbf{AM}[2t(n)]$ and $\mathbf{IP}[2t(n)]$ could be avoided.

2 Details

In this section, we give some detailed comments on the different parts of the survey.

2.1 Interactive proof systems (Section 2)

When the framework for interactive proofs is introduced, it is mentioned that the other standard model is based on Turing machines that communicate using several shared tapes. It is claimed that the two models "should be identical in all respects" for the purpose of language recognition. The formulation ("*should*") suggests that the equivalence between the models is an open question. It this intentional?

The description of the protocol just before Definition 2.1 would be easier to read if it would explicitly be mentioned that y_i is the message sent from V to P, and z_i is the message sent from P to V during round i. This could simply be achieved by writing "The message sent... is $y_i = V(x, r, s_i)$. Conversely, the message sent... is $z_i = P(x, s_i \# y_i)$.".

In Definition 2.1, a 'w' appears in $V(w, r, s_k)$. Is this intentional?

2.2 Examples of interactive proof systems (Section 3)

The first paragraph is a bit unclear ("... *the* class of problems..." (emphasis added by the reviewer)).

Just after the problems Graph Isomorphism and Graph Nonisomorphism have been introduced, it is claimed that the existence of an interactive proof for a language does not immediately imply that there is an interactive proof for the complement language. Since $\mathbf{IP} = \mathbf{PSPACE}$ and $\mathbf{coPSPACE} = \mathbf{PSPACE}$, $L \in \mathbf{IP}$ actually implies that the complement of L is in $\mathbf{IP}$. (Of course, the implication is not immediate, since the known proofs that $\mathbf{IP} = \mathbf{PSPACE}$ are non-trivial.)

The first step in the protocol for Graph Isomorphism is actually unnecessary, since both parties know the graphs. Thus, the first step is a completely dummy step.

In the interactive protocol for Graph Nonisomorphism, the (technical) reason why the first step is repeated twice might be unclear (at least for an careless reader).

2.3 Arthur-Merlin games (Section 5)

The formulation in the first paragraph might be a little unclear. The crucial fact is that Arthur is not only restricted to sending random strings, but is restricted to send a prefix of his random string.

In Definition 5.1, is Arthur completely deterministic, except for the random coins he sends to Merlin, or is the predicate computed by Arthur in order to accept or reject allowed to be probabilistic? As stated, the definition allows the predicate to be probabilistic. Is this essential? In that case, why?

2.4 The Chernoff Bound and the Majority Vote Lemma (Appendix A)

There is a nasty typo in the Chernoff Bound. The expression should be

$$\Pr\left(\sum_{i=1}^{n} x_i \geq (1+\theta)pn\right) \leq e^{pn(\theta-(1+\theta)\ln(1+\theta))}.$$

Similarly, $g(\theta)$ in the proof of the Majority Vote Lemma should be $g(\theta) = \theta - (1+\theta)\ln(1+\theta)$.

3 Further discussion

Finally, we give some suggestions for further discussions.

3.1 Consequences for cryptography

What are the consequences of all this for cryptographic protocols?

Interactive proofs can be seen as prototypes for cryptographic protocols (in cryptographic protocols, we usually have extra requirements like zero knowledge and robustness against nasty adversaries). Since any interactive proof with a constant number of rounds can be converted to a constant round public-coin protocol (by the $\mathbf{IP}[2t(n)] \subseteq \mathbf{AM}[2t(n)+2]$ result), and furthermore be speeded up to only a few rounds (by the speedup theorem), this suggest similar results applies to various cryptographic protocols. Under what condition is this speedup possible, and what is the minimum number of rounds needed? What is the true role of randomness in the protocols?

Put another way, what additional requirements can we add to an interactive proof and still be able to apply the "private coins equals public coins" theorem or the speedup theorem?

(For zero knowledge, the privacy of the verifier's coins is very essential; see [1].)

3.2 On the power of the parties

How powerful should the prover be? We know that polynomial space computation is enough for the prover, but what happens if we restrict the prover in various ways? Similarly, what happens it we increase the computational abilities of the verifier?

3.3 Classical proofs

In the survey, it is argued that $\mathbf{NP}$ can be seen as the complexity-theoretic version of classical proofs. Is this actually the case? What are the "computational abilities" of a verifier? Why is the prover computationally more powerful?

In classical proofs, the verifier (for example, the reader of a paper) is often required to supply parts of the proof (for example, statements like "As the reader can verify..."). Is the verifier in any sense restricted to polynomial time computation? For example, a skilled mathematician should be able to do fairly sophisticated verification. In addition, a scientific paper usually have several collaborating verifiers (editors of journals, conference attenders, students in lectures, ...). Similarly, a mathematical statement might have several independent or collaborating provers (for example, in the form of several published proofs). Can **NP** really be considered to be the complexity theoretic analogue of classical proofs? Is not different forms of interaction an essential part of the long verification process of new scientific results?

These questions, of course, go way beyond the scope of the survey, but they might be interesting topics for philosophical discussions.

What is a proof, really? And why should I buy it?

References

[1] O. Goldreich and H. Krawczyk. On the composition of zero-knowledge proof systems. *SIAM Journal of Computing*, 25(1):169–192, 1996.

[2] Petteri Kaski. T-79.510 Special Course on Cryptology / Zero Knowledge: Rudiments of interactive proof systems. Helsinki University of Technology, 26 September 2001.