Course "Proofs and Computers", JASS'06

Introducing IP, AM, MA

Florian Zuleger

Fakultät für Informatik
TU München

March 30, 2006

We can view NP as a proof system. For each language $L \in$ NP there exists a polynomial-time recognizable relation R_L such that:

$$L = \{x | \exists y : s.t.(x, y) \in R_L\}$$

and $(x, y) \in R_L$ only if $|y| \leq poly(|x|)$.

A good proof system must have the following properties:

1. The verifier strategy is efficient (polynomial-time in the NP case)

A good proof system must have the following properties:

1. The verifier strategy is efficient (polynomial-time in the NP case)
2. Correctness requirements:

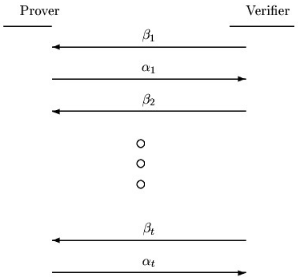
A good proof system must have the following properties:

1. The verifier strategy is efficient (polynomial-time in the NP case)
2. Correctness requirements:
   - **Completeness**: For a true assertion, there is a convincing proof strategy (in the case of NP, if $x \in L$ the a witness y exists).

A good proof system must have the following properties:

1. The verifier strategy is efficient (polynomial-time in the NP case)
2. Correctness requirements:
   - **Completeness**: For a true assertion, there is a convincing proof strategy (in the case of NP, if $x \in L$ the a witness y exists).
   - **Soundness**: For a false assertion, no convincing proof strategy exists (in the case of NP, if $x \notin L$ then no witness y exists).

Now we generalize the requirements from a proof system, adding interaction and randomness.
An interactive proof is sequence of questions and answers between the prover and the verifier.



At the end of the interaction, the verifier decides based the knowledge he acquired in the process whether the claim is true or false.

Definition 1

(interactive proof systems:) An interactive proof system for a language L is a two-party game between a verifier and a prover that interact on a common input in a way satisfying the following properties:

Definition 1

(interactive proof systems:) An interactive proof system for a language L is a two-party game between a verifier and a prover that interact on a common input in a way satisfying the following properties:

1. The verifier strategy is a probabilistic polynomial-time procedure (where time is measured in terms of the length of the common input)

Definition 1

(interactive proof systems:) An interactive proof system for a language L is a two-party game between a verifier and a prover that interact on a common input in a way satisfying the following properties:

1. The verifier strategy is a probabilistic polynomial-time procedure (where time is measured in terms of the length of the common input)
2. Correctness requirements:

Definition 1

(interactive proof systems:) An interactive proof system for a language L is a two-party game between a verifier and a prover that interact on a common input in a way satisfying the following properties:

1. The verifier strategy is a probabilistic polynomial-time procedure (where time is measured in terms of the length of the common input)
2. Correctness requirements:
   - **Completeness**: There exists a prover strategy P, such that for every $x \in L$, when interacting on the common input x, the prover P convinces the verifier with probability at least $\frac{2}{3}$.

Definition 1

(interactive proof systems:) An interactive proof system for a language L is a two-party game between a verifier and a prover that interact on a common input in a way satisfying the following properties:

1. The verifier strategy is a probabilistic polynomial-time procedure (where time is measured in terms of the length of the common input)
2. Correctness requirements:
   - **Completeness**: There exists a prover strategy P, such that for every $x \in L$, when interacting on the common input x, the prover P convinces the verifier with probability at least $\frac{2}{3}$.
   - **Soundness**: For a false assertion, no convincing proof strategy exists (in the case of NP, if $x \notin L$ then no witness y exists).

Definition 2

(The IP Hierachy:) The complexity class IP consists of all languages having an interactive proof system.
We call the number of message exchanges (a question and an answer) between the two parties, the number of rounds in the system. After a certain number of rounds the verifier decides whether to accept or reject.
For every integer function r(.), the complexity class IP(r(.)) consists of all the languages that have an interactive proof system in which, on common input x, at most $r(|x|)$ rounds are used.
If we denote by **poly** the set of all integer polynomial functions, then $IP = IP(poly)$.

Comments

- Clearly, $NP \subseteq IP(1)$.
  Also, $BPP = IP(0)$.

- Clearly, $NP \subseteq IP(1)$.
  Also, $BPP = IP(0)$.
- The number of rounds in IP cannot be more than a polynomial in the length of the common input.

- Clearly, $NP \subseteq IP(1)$.
  Also, $BPP = IP(0)$.
- The number of rounds in IP cannot be more than a polynomial in the length of the common input.
- The length of the messages exchanged cannot be more than a polynomial in the length of the common input.

- Clearly, $NP \subseteq IP(1)$.
  Also, $BPP = IP(0)$.
- The number of rounds in IP cannot be more than a polynomial in the length of the common input.
- The length of the messages exchanged cannot be more than a polynomial in the length of the common input.

- Clearly, $NP \subseteq IP(1)$.
  Also, $BPP = IP(0)$.
- The number of rounds in IP cannot be more than a polynomial in the length of the common input.
- The length of the messages exchanged cannot be more than a polynomial in the length of the common input.

- **Claim 3**
  Any language that has an interactive proof system, has one that achieves error probability of at most $2^{-p(.)}$ for any polynomial p(.).

Proof.

Using Chernoff's Bound:

$$Pr[z < (1-\delta)E(z)] < e^{-\frac{\delta^2 E(z)}{2}}$$

We choose $k = O(p(.))$ and $\delta = \frac{1}{4}$ and note that $E(z) = \frac{2}{3}k$ (so that $\frac{3}{4} \cdot \frac{2}{3} = \frac{1}{2}$) to get:

$$Pr[z < (1 - \frac{1}{2}k] < 2^{-p(.)}$$

□

Introducing both interaction and randomness in the IP class is essential:

- By adding interaction only, the interactive proof systems collapse to NP-proof systems.

Introducing both interaction and randomness in the IP class is essential:

- By adding interaction only, the interactive proof systems collapse to NP-proof systems.
- By adding randomness only, we get a proof system in which the prover sends a witness and the verifier can run a BPP algorithm for checking its validity. We obtain the class Merlin-Arthur game - MA.

Example 4 (Graph Non-Isomorphism(GNI))

Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are called *iosmorphic* (denoted $G_1 \cong G_2$) if there exists a 1-1 and onto mapping $\pi : V_1 \rightarrow V_2$ such that $(u, v) \in E_1 \Leftrightarrow (\pi(u), \pi(v)) \in E_2$. The mapping π, if existing, is called an *isomporhism* between the graphs. If no such mapping exists then the graphs are *non-isomophic* (denoted $G_1 \ncong G_2$).

GNI is the language containing all pairs of non-isomorphic graphs. Formally:

$$GNI = \{(G_1, G_2) : G_1 \ncong G_2\}$$

An interactive proof system for GNI:

- G_1 and G_2 are given as input to the verifier and the prover. Assume without loss of generality that $V_1 = V_2 = \{1, 2, \ldots, n\}$.

An interactive proof system for GNI:

- G_1 and G_2 are given as input to the verifier and the prover. Assume without loss of generality that $V_1 = V_2 = \{1, 2, \ldots, n\}$.
- The verifier chooses $i \in_R \{1, 2\}$ and $\pi \in_R S_n$ (S_n is the group of all permutations on $\{1, 2, \ldots, n\}$). He applies the mapping π on the graph G_i to obtain a graph H

  $$H = (\{1, 2, \ldots, n\}, E_H) \text{ where } E_H = \{(\pi(u), \pi(v)) : (u, v) \in E_i\}$$

  and sends the graph H to the prover.

An interactive proof system for GNI:

- G_1 and G_2 are given as input to the verifier and the prover. Assume without loss of generality that $V_1 = V_2 = \{1, 2, \ldots, n\}$.
- The verifier chooses $i \in_R \{1, 2\}$ and $\pi \in_R S_n$ (S_n is the group of all permutations on $\{1, 2, \ldots, n\}$). He applies the mapping π on the graph G_i to obtain a graph H

  $$H = (\{1, 2, \ldots, n\}, E_H) \text{ where } E_H = \{(\pi(u), \pi(v)) : (u, v) \in E_i\}$$

  and sends the graph H to the prover.
- The prover sends $j \in \{1, 2\}$ to the verifier.

An interactive proof system for GNI:

- G_1 and G_2 are given as input to the verifier and the prover. Assume without loss of generality that $V_1 = V_2 = \{1, 2, \ldots, n\}$.
- The verifier chooses $i \in_R \{1, 2\}$ and $\pi \in_R S_n$ (S_n is the group of all permutations on $\{1, 2, \ldots, n\}$). He applies the mapping π on the graph G_i to obtain a graph H

  $$H = (\{1, 2, \ldots, n\}, E_H) \text{ where } E_H = \{(\pi(u), \pi(v)) : (u, v) \in E_i\}$$

  and sends the graph H to the prover.
- The prover sends $j \in \{1, 2\}$ to the verifier.
- The verifier accepts iff $j = i$.

- **Remark:** ISOMORPHISM is not known to be in P, but of course it is in NP (guessing the right permutation and then checking the isomorphism in polynomial time), whereas GNI is not known to be in NP.

- **Remark:** ISOMORPHISM is not known to be in P, but of course it is in NP (guessing the right permutation and then checking the isomorphism in polynomial time), whereas GNI is not known to be in NP.
- **Remark:** We state that the secrecy of the outcome of the coin tosses is essential to this protocol.

Definition 5

(public-coin interactive proofs - AM:) Public coin proof systems (known also as Arthur-Merlin games) are a special case of interactive proof systems, in which, at each round, **the verifier can only toss coins, and send their outcome to the prover**. After a certain number of rounds the verifier decides **deterministically** whether to accept or reject.
For every integer function r(.), the complexity class AM(r(.)) consists of all the languages that have an Arthur-Merlin proof system in which, on common input x, at most $r(|x|)$ rounds are used.
Denote $AM = AM(1)$.

Surprisingly it was shown Arthur-Merlin games and the general interactive proof systems are essentially equivalent:

Theorem 6

(Relating IP(.) to AM(.)):

$$\forall r(.) : IP(r(.)) \subseteq AM(r(.) + 1)$$

The following theorem shows that power of AM(r(.)) is invariant under a linear change in the number of rounds:

Theorem 7

(Linear Speed-up Theorem):

$$\forall r(.) \geq 2 : AM(2r(.)) = AM(r(.))$$

Combing the two las theorems we get:

Corollary 8

$$\forall r(.) \geq 2 : IP(2r(.)) = IP(r(.))$$

Combing the two las theorems we get:

Corollary 8

$$\forall r(.) \geq 2 : IP(2r(.)) = IP(r(.))$$

Corollary 9

(Collapse of constant-round IP to one-round AM):

$$IP(O(1)) = AM(1)$$

Combing the two las theorems we get:

Corollary 8

$$\forall r(.) \geq 2 : IP(2r(.)) = IP(r(.))$$

Corollary 9

(Collapse of constant-round IP to one-round AM):

$$IP(O(1)) = AM(1)$$

Corollary 10

(Relating MA to AM)

$$MA \subseteq AM$$

Theorem 11
(Relating MA to PP):

$$MA \subseteq PP$$

Proof. Let $L \in MA$. Thus there are a polynomial p and a polynomial-time Turing machine Q such that:

$$x \in L \Rightarrow \exists s \in \{0,1\}^{p(|x|)} : Pr[Q(x,r,x)] > \frac{2}{3}$$

$$x \notin L \Rightarrow \forall s \in \{0,1\}^{p(|x|)} : Pr[Q(x,r,x)] < \frac{1}{3}$$

where probability is taken over uniform distribution in $\{0,1\}^{p(|x|)}$.

Using standard amplification we can construct a new polynomial p_1 and a new polynomial-time machine Q_1 such that

$$x \in L \Rightarrow \exists s \in \{0,1\}^{p(|x|)} : Pr[Q_1(x,r,s)] > 1 - 4^{-p(|x|)}$$

$$x \notin L \Rightarrow \forall s \in \{0,1\}^{p(|x|)} : Pr[Q_1(x,r,s)] < 4^{-p(|x|)}$$

where probability is taken over uniform distribution in $\{0,1\}^{p_1(|x|)}$. Consider now the uniform distribution on pairs $< r, s > \in \{0,1\}^{p(|x|)+p_1(|x|)}$. We have

$$x \in L \Rightarrow \exists Pr[Q_1(x,r,s)] > 2^{-p(|x|)}(1 - 4^{-p(|x|)}) > 4^{-p(|x|)}$$

$$x \notin L \Rightarrow Pr[Q_1(x,r,s)] < 4^{-p(|x|)}$$

This is equivalent to $L \in PP$. $\square$

What if we require Perfect Completeness, i.e., convincing the verifier with probability 1?

Theorem 12

If a language has an interactive proof system then it has one with perfect completeness.

We will show that given a public coin proof system we can construct a perfect completeness public coin proof system.
We define:
$AM^0(r(.)) = \{L|\ L$ has a perfect completeness
$r(.)$ round public coin proof system$\}$

We will show:

Lemma 13

If L has a public coin proof system then it has one with perfect completeness

$$AM(r(.)) \subseteq AM^0(r(.)+1)$$

Proof.

- Assume that the Arthur-Merlin proof system consists of t rounds.

Proof.

- Assume that the Arthur-Merlin proof system consists of t rounds.
- Assume that Arthur sends the same number of coins m in each round.

Proof.

- Assume that the Arthur-Merlin proof system consists of t rounds.
- Assume that Arthur sends the same number of coins m in each round.
- Also assume that the completeness and soundness error probabilities of the proof system are at most $\frac{1}{3tm}$. This is obtained using standard amplification.

Proof.

- Assume that the Arthur-Merlin proof system consists of t rounds.
- Assume that Arthur sends the same number of coins m in each round.
- Also assume that the completeness and soundness error probabilities of the proof system are at most $\frac{1}{3tm}$. This is obtained using standard amplification.
- We denote the messages sent by Arthur (the verifier) $r_1, \ldots, r_t$ and the messages sent by Merlin (the prover) $\alpha_1, \ldots, \alpha_t$.

Proof.

- Assume that the Arthur-Merlin proof system consists of t rounds.
- Assume that Arthur sends the same number of coins m in each round.
- Also assume that the completeness and soundness error probabilities of the proof system are at most $\frac{1}{3tm}$. This is obtained using standard amplification.
- We denote the messages sent by Arthur (the verifier) $r_1, \ldots, r_t$ and the messages sent by Merlin (the prover) $\alpha_1, \ldots, \alpha_t$.
- Denote by $< P, V >_x (r_1, \ldots, r_t)$ the outcome of the game on common input x between the optimal prover P and the verifier V.

- We construct a new proof system with perfect completeness, in which Arthur and Merlin play *tm* games simultaneously.

- We construct a new proof system with perfect completeness, in which Arthur and Merlin play tm games simultaneously.
- Each game is like the original game except that the random coins are shifted by a fixed amount.

- We construct a new proof system with perfect completeness, in which Arthur and Merlin play tm games simultaneously.
- Each game is like the original game except that the random coins are shifted by a fixed amount.
- Formally, we add an additional round at the beginning in which Merlin sends the tm shifts $S^1, \dots, S^{tm}$ where $S^i = (S^i_1, \dots, S^i_t), S^i_j \in \{0,1\}^m$ to Arthur.

- For game i and round j, Merlin considers the random coins to be $r_j \oplus S_j^i$ and sends as a message α_j^i where α_j^i is computed according to $(r_1 \oplus S_1^i, \ldots, r_t \oplus S_t^i)$.

- For game i and round j, Merlin considers the random coins to be $r_j \oplus S_j^i$ and sends as a message α_j^i where α_j^i is computed according to $(r_1 \oplus S_1^i, \dots, r_t \oplus S_t^i)$.
- The entire message in round j is $\alpha_j^1, \dots, \alpha_j^{tm}$.

- For game i and round j, Merlin considers the random coins to be $r_j \oplus S^i_j$ and sends as a message α^i_j where α^i_j is computed according to $(r_1 \oplus S^i_1, \ldots, r_t \oplus S^i_t)$.
- The entire message in round j is $\alpha^1_j, \ldots, \alpha^{tm}_j$.
- At the end of the protocol Arthur accepts if at least one out of the tm games is accepting.

- In order to show perfect completeness we will show that for every $x \in L$ there exists $S^1, \dots, S^{tm}$ such that for all $r_1, \dots, r_t$ at least one of the games is accepting.

- In order to show perfect completeness we will show that for every $x \in L$ there exists $S^1, \ldots, S^{tm}$ such that for all $r_1, \ldots, r_t$ at least one of the games is accepting.
- We use a probabilistic argument to show that the complementary event occurs with probability strictly smaller than 1.

$$Pr_{S^1,\ldots,S^{tm}}[\exists r_1,\ldots,r_t \bigwedge_{i=1}^{tm}(< P, V >_x (r_1 \oplus S^i_1,\ldots,r_t \oplus S^i_t) = 0)]$$

$$\leq_{(1)} \sum_{r_1,\ldots,r_t \in \{0,1\}^m} Pr_{S^1,\ldots,S^{tm}}[\bigwedge_{i=1}^{tm}(< P, V >_x (r_1 \oplus S^i_1,\ldots,r_t \oplus S^i_t) = 0)]$$

$$\leq_{(2)} 2^{tm} \cdot (\frac{1}{3tm})^{tm} < 1$$

Inequality (1) is obtained using the union bound. Inequality (2) is due to the fact that the $r_j \oplus S^i_j$ are independent random variables so the results of the games are independent, and that the optimal prover fails to convince the verifier on a true assertion with probability at most $\frac{1}{3tm}$.

We still have to show that the proof system suggested satisfies the soundness requirement. We show that for every $x \notin L$ and for any prover strategy $P^\star$ and choices of shifts $S^1, \dots, S^{tm}$ the probability that one or more of the tm games is accepting is at most $\frac{1}{3}$.

$$Pr_{r_1,\ldots,r_t}[\bigvee_{i=1}^{tm}(< P, V >_x (r_1 \oplus S_1^i, \ldots, r_t \oplus S_t^i) = 1)]$$

$$\leq_{(1)} \sum_{i=1}^{tm} Pr_{r_1,\ldots,r_t}[< P^\star, V >_x (r_1 \oplus S_1^i, \ldots, r_t \oplus S_t^i) = 1)]$$

$$\leq_{(2)} \sum_{i=1}^{tm} \frac{1}{3tm} = \frac{1}{3}$$

Inequality (1) is obtained using the union bound. Inequality (2) is due to the fact that any prover has probability of at most $\frac{1}{3tm}$ of success for a single game. $\square$

Unlike the last theorem, requiring *perfect soundness* reduces the model to an NP-proof system.

Proposition 14

If a lanuage L has an interactive proof system with perfect soundness then $L \in NP$.

Remark: (This is an alternative argument for interactive proof systems collapsing to NP without randomness. This is due to the fact that perfect soundness is equivalent to a deterministic verifier.)

We shall conclude this paper with a very interesting protocol that uses interactive proofs and cryptography.
Suppose that Alice is a girl with superintellectual abilities capable to solve NP-problems.

And suppose that Bob - an ordinary guy, but a good friend - is only able to compute problems in P.

Alice knows a 3-coloring of a large graph $G = (V, E)$ and wants to convince Bob that she has a coloring of G without telling him the coloring.

What is required here is a zero knowledge proof, that is, an interactive protocol at the end of which Bob is convinced that with very high probability Alice has a legal 3-coloring of G, but has no clue about the actual 3-coloring.

Alice's coloring is $\chi : V \mapsto \{00, 11, 01\}$. The protocol proceeds in rounds. At each round, Alice carries out the following steps:

- She generates a random permutation π of the three colors.

Alice's coloring is $\chi : V \mapsto \{00, 11, 01\}$. The protocol proceeds in rounds. At each round, Alice carries out the following steps:

- She generates a random permutation π of the three colors.
- She generates $|V|$ RSA public-private key pairs, (p_i, q_i, d_i, e_i), one for each node $i \in V$.

Alice's coloring is $\chi : V \mapsto \{00, 11, 01\}$. The protocol proceeds in rounds. At each round, Alice carries out the following steps:

- She generates a random permutation π of the three colors.
- She generates $|V|$ RSA public-private key pairs, (p_i, q_i, d_i, e_i), one for each node $i \in V$.
- For each node i she computes the probabilistic encoding (y_i, y_i'), according the $j$the RSA system, of the color $\pi(\chi(i))$.

- Supppose that $b_i b'_i$ are the two bits of $\pi(\chi(i))$; then $y_i = (2x_i + b_i)^{e_i} mod p_i q_i$ and $y'_i = (2x'_i + b'_i)^{e_i} mod p_i q_i$, where x_i and x'_i are random integers no greater than $\frac{pq}{2}$.

- Supppose that $b_i b'_i$ are the two bits of $\pi(\chi(i))$; then $y_i = (2x_i + b_i)^{e_i} mod p_i q_i$ and $y'_i = (2x'_i + b'_i)^{e_i} mod p_i q_i$, where x_i and x'_i are random integers no greater than $\frac{pq}{2}$.
- All these computations are private to Alice. Alice reveals to Bob the integers $(e_i, p_i q_i, y_i, y'_i)$ for each node $i \in V$. That is, the public part of the RSA systems, and the encrypted colors.

Bob's turn:

- Bob picks at random an edge $[i, j] \in E$, and inquires whether its endpoints have a different color.

Bob's turn:

- Bob picks at random an edge $[i, j] \in E$, and inquires whether its endpoints have a different color.
- Alice then reveals to Bob the secret keys d_i and d_j of the endpoints, allowing Bob to compute $b_i = y_i^{e_i} mod 2$, and similarly for b'_i,b_j and b'_j.

Bob's turn:

- Bob picks at random an edge $[i, j] \in E$, and inquires whether its endpoints have a different color.
- Alice then reveals to Bob the secret keys d_i and d_j of the endpoints, allowing Bob to compute $b_i = y_i^{e_i} mod 2$, and similarly for b'_i,b_j and b'_j.
- He checks that indeed $b_i b'_i \neq b_j b'_j$.

Alice and Bob repeat $k|E|$ times, where k is a parameter representing the desired reliability of the protocol.

- If Alice has a legal coloring of G, all inquiries of Bob will be satisfied.

Alice and Bob repeat $k|E|$ times, where k is a parameter representing the desired reliability of the protocol.

- If Alice has a legal coloring of G, all inquiries of Bob will be satisfied.
- If she has no legal coloring, then necessarily at each round there is an edge $[i, j] \in E$ such that $\chi(i) = \chi(j)$.

Alice and Bob repeat $k|E|$ times, where k is a parameter representing the desired reliability of the protocol.

- If Alice has a legal coloring of G, all inquiries of Bob will be satisfied.
- If she has no legal coloring, then necessarily at each round there is an edge $[i,j] \in E$ such that $\chi(i) = \chi(j)$.
- At each round Bob has a probability of at least $\frac{1}{|E|}$ if discovering that edge.

Alice and Bob repeat $k|E|$ times, where k is a parameter representing the desired reliability of the protocol.

- If Alice has a legal coloring of G, all inquiries of Bob will be satisfied.
- If she has no legal coloring, then necessarily at each round there is an edge $[i, j] \in E$ such that $\chi(i) = \chi(j)$.
- At each round Bob has a probability of at least $\frac{1}{|E|}$ if discovering that edge.
- After $k|E|$ rounds, the probability of Bob finding out that Alice has no legal coloring is at least $1 - e^{-k}$.

- What is remarkable about this protocol is that Bob has learned nothing about Alice's coloring of G in the process.

- What is remarkable about this protocol is that Bob has learned nothing about Alice's coloring of G in the process.
- As a final note the zero knowledge protocol just described works for 3-COLORING, an NP-complete problem. Using reductions, it is possible to conclude all problems in NP have zero-knowledge proofs.