The actual protocol

Prover
\[ p_1(x) = \sum_{x_2, \ldots, x_n \in \{0,1\}} p_1(x, x_2, \ldots, x_n) z \]
\[ p_2(x) = \sum_{x_3, \ldots, x_n \in \{0,1\}} p_2(x_1, x_2, x_3, \ldots, x_n) \]
\[ p_3(x) = \sum_{x_4, \ldots, x_n \in \{0,1\}} p_3(x_1, x_2, x_3, x_4, \ldots, x_n) \]
\[ p_n(x) = \sum_{x_n \in \{0,1\}} p_n(x_1, x_2, \ldots, x_n) \]

Verifier
\[ p_1(0)+p_1(1)=k? \]
\[ p_2(0)+p_2(1)=p(z_1)? \]
\[ p_3(0)+p_3(1)=p(z_2)? \]
\[ p_n(0)+p_n(1)=p(z_n)? \]

May 13, 2021 CS151 Lecture 14

Analysis of protocol

- Completeness:
  - if \((\varphi, k) \in L\) then honest prover on previous slide will always cause verifier to accept

- Soundness:
  - let \(p_i(x)\) be the correct polynomials
  - let \(p_i^*(x)\) be the polynomials sent by (cheating) prover
  - \((\varphi, k) \notin L \Rightarrow p_1(0) + p_1(1) \neq k\)
  - either \(p_1^*(0) + p_1^*(1) \neq k\) (and V rejects)
  - or \(p_1^* \neq p_1\)
  - assume \((p_{i+1}(0)+p_{i+1}(1)=p_{i+1}(z_i)) \neq p_{i+1}(z_i)\)
  - either \(p_{i+1}^*(0) + p_{i+1}^*(1) \neq p_{i+1}(z_i)\) (and V rejects)
  - or \(p_{i+1}^* \neq p_{i+1}\)

- Soundness (continued):
  - if verifier does not reject, there must be some \(i\) for which:
    - \(p_i^* \neq p_i\) and yet \(p_i^*(z_i) = p_i(z_i)\)
    - for each \(i\), probability is \(\leq |\varphi|/2^n\)
    - union bound: probability that there exists an \(i\) for which the bad event occurs is \(\leq n|\varphi|/2^n \leq \text{poly}(n)/2^n < 1/3\)

Analysis of protocol

- Conclude: \(L = \{(\varphi, k) : \text{CNF } \varphi \text{ has exactly } k \text{ satisfying assignments}\}\) is in \(\text{IP}\)

- \(L\) is \(\text{coNP}\)-hard, so \(\text{coNP} \subseteq \text{IP}\)

- Question remains:
  - \(\text{NP}, \text{coNP} \subseteq \text{IP}\). Potentially larger. How much larger?
**Theorem:** (Shamir) $\text{IP} = \text{PSPACE}$

– Note: $\text{IP} \subseteq \text{PSPACE}$
  * enumerate all possible interactions, explicitly calculate acceptance probability
  * interaction extremely powerful!
  * An implication: you can interact with master player of Generalized Geography and determine if she can win from the current configuration even if you do not have the power to compute optimal moves!

**protocol for QSAT**

– arithmetization step produces arithmetic expression $p_\varphi$:
  * $(\exists x_i) \varphi \rightarrow \Sigma x_i = 0, 1 \ p_\varphi$
  * $(\forall x_i) \varphi \rightarrow \Pi x_i = 0, 1 \ p_\varphi$

– start with QSAT formula in special form (“simple”)
  * no occurrence of $x_i$ separated by more than one “$\exists$” from point of quantification

**The QSAT protocol**

<table>
<thead>
<tr>
<th>Prover</th>
<th>input: $\varphi$</th>
<th>Verifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k, q, p(x)$</td>
<td>$p_1(x)$: remove outer $\Sigma$ or $\Pi$ from $p_2$</td>
<td>$z_1$</td>
</tr>
<tr>
<td></td>
<td>$p_1(x)$: remove outer $\Sigma$ or $\Pi$ from $p_0$</td>
<td>$z_2$</td>
</tr>
<tr>
<td></td>
<td>$p_2(x)$</td>
<td>$z_3$</td>
</tr>
<tr>
<td></td>
<td>$p_3(x)$: remove outer $\Sigma$ or $\Pi$ from $p_1(x)$</td>
<td>$z_4$</td>
</tr>
<tr>
<td></td>
<td>$p_4(x)$</td>
<td>$z_5$</td>
</tr>
<tr>
<td></td>
<td>$p_n(x)$</td>
<td>$z_n$</td>
</tr>
</tbody>
</table>

Analysis of the QSAT protocol

• Completeness:
  – if $\varphi \in \text{QSAT}$ then honest prover on previous slide will always cause verifier to accept
Analysis of the QSAT protocol

- Soundness:
  - let $p_i(x)$ be the correct polynomials
  - let $p_i^*(x)$ be the polynomials sent by (cheating) prover
  - $\phi \not\in \text{QSAT} \Rightarrow 0 = p_1(0) + p_1(1) = k$
  - either $p_1^*(0) + p_1^*(1) = k$ (and $V$ rejects)
  - or $p_i^* \neq p_i \Rightarrow Pr_z[p_i^*(z_1) = p_i(z_1)] \leq 2|\phi|/2^n$
  - assume $p_i+1(0) + p_i+1(1) = p_i(z_i) \neq p_i^*(z_i)$
  - either $p_i+1^*(0) + p_i+1^*(1) = p_i^*(z_i)$ (and $V$ rejects)
  - or $p_i+1^* \neq p_i+1 \Rightarrow Pr_z[p_i^*(z_i) = p_i(z_i)] \leq 2|\phi|/2^n$

Example

- Papadimitriou – pp. 475-480

$\phi = \forall x \exists y (x \land y) \land \forall z ((x \land z) \lor (y \land \neg z)) \land \exists w (z \lor (y \land \neg w))$

$p_\phi = \prod_{x=0,1} \sum_{y=0,1} [(x + y) \cdot \prod_{z=0,1} [(x \lor z) \lor \neg \sum_{w=0,1} (z \lor (y \lor \neg w))]$

$p_\phi = 96$ but $V$ doesn’t know that yet!

Example

$\phi = \forall x \exists y (x \lor y) \land \forall z ((x \land z) \lor (y \land \neg z)) \land \exists w (z \lor (y \land \neg w))$

$p_\phi = \prod_{x=0,1} \sum_{y=0,1} [(x + y) \cdot \prod_{z=0,1} [(x \lor z) \lor \neg \sum_{w=0,1} (z \lor (y \lor \neg w))]$

Round 2: (prover claims this = 6)
- verifier removes outermost "|"; sends $p_2(x) = 2y^3 + y^2 + 3y$
- verifier checks:
  - $p_2(0) + p_2(1) = 0 + 6 = 6 \equiv 6$ (mod 13)
- verifier picks randomly: $z_2 = 3$

Analysis of protocol

- Soundness (continued):
  - if verifier does not reject, there must be some $i$ for which:
    - $p_i^* \neq p_i$ and yet $p_i^*(z_i) = p_i(z_i)$
    - for each $i$, probability is $\leq 2|\phi|/2^n$
  - union bound: probability that there exists an $i$ for which the bad event occurs is $\leq 2n|\phi|/2^n \leq \text{poly}(n)/2^n \ll 1/3$

- Conclude: QSAT is in IP
Example

\[ \varphi = \forall x \exists y (x y) \land \forall z ((x z) v (y \land \neg z)) v \exists w (z v (y \land \neg w)) \]

\[ p_\varphi = \prod_{x=0,1} \sum_{y=0,1} [(x + y) \times (z + y(1-w))] \]

Round 3: (prover claims this = 7)
- everyone agrees expression = 12*(…)
- prover removes outermost \(\prod\); sends 
  \[ p_3(z) = 8z + 6 \]
- verifier checks:
  \[ p_3(0) \times p_3(1) = (6)(14) = 84; 12*84 \equiv 7 \pmod{13} \]
- verifier picks randomly: \( z_3 = 2 \)

Final check:
\[ 12 * [9*7 + 3(1-7)] + \sum_{w=0,1} (7 + 3(1-w)) \]

Arthur-Merlin Games

• **IP** permits verifier to keep coin-flips **private**
  - necessary feature?
  - GNI protocol breaks without it

• **Arthur-Merlin game**: interactive protocol in which coin-flips are **public**
  - Arthur (verifier) may as well just send results of coin-flips and ask Merlin (prover) to perform any computation Arthur would have done

• Clearly **Arthur-Merlin \(\subseteq\) IP**
  - “private coins are at least as powerful as public coins”

• Proof that **IP = PSPACE** actually shows
  \[ \text{PSPACE} \subseteq \text{Arthur-Merlin} \subseteq \text{IP} = \text{PSPACE} \]
  - “public coins are at least as powerful as private coins”!
**Theorem**: for all constant $k \geq 2$ 
\[ AM[k] = AM[2] \]

- **Proof**:
  - we show $MA[2] \subseteq AM[2]$
  - implies can move all of Arthur’s messages to beginning of interaction:
    
    \[ AMAMAM...AM = AAMAM...AM \]

- **Two important classes**:
  - $MA = MA[2]$
  - $AM = AM[2]$

- definitions without reference to interaction:
  - $L \in MA$ iff 3 poly-time language $R$
    \[ x \in L \Rightarrow \exists m Pr[(x, m, r) \in R] = 1 \]
    \[ x \notin L \Rightarrow \forall m Pr[(x, m, r) \in R] \leq \epsilon \]
  - $L \in AM$ iff 3 poly-time language $R$
    \[ x \in L \Rightarrow Pr[\exists m (x, m, r) \in R] = 1 \]
    \[ x \notin L \Rightarrow Pr[\exists m (x, m, r) \in R] \leq \frac{\epsilon}{2} \]

- Relation to other complexity classes:
  - both contain $NP$ (can elect to not use randomness)
  - both contained in $\Pi_2$, $L \in \Pi_2$ iff $R \in P$ for which:
    \[ x \in L \Rightarrow Pr[\exists m (x, m, r) \in R] = 1 \]
    \[ x \notin L \Rightarrow Pr[\exists m (x, m, r) \in R] < 1 \]
  - so clear that $AM \subseteq \Pi_2$
  - know that $MA \subseteq AM$
**MA and AM**

**Theorem:** $\text{coNP} \subseteq \text{AM} \Rightarrow \text{PH} = \text{AM}$.

**Proof:**
- suffices to show $\Sigma_2 \subseteq \text{AM}$ (and use $\text{AM} \subseteq \Pi_2$)
- $L \in \Sigma_2$ iff $\exists$ poly-time language $R$
  
  $x \in L \Rightarrow \exists y \forall z (x, y, z) \in R$
  
  $x \notin L \Rightarrow \forall y \exists z (x, y, z) \notin R$

- Merlin sends $y$
- 1 AM exchange decides $\text{coNP}$ query: $\forall z (x, y, z) \in R$?
- 3 rounds; in $\text{AM}$

**MA and AM**

- We know $\text{Arthur-Merlin} = \text{IP}$.
  
  - “public coins = private coins”

  **Theorem (GS):** $\text{IP}[k] \subseteq \text{AM}[O(k)]$
  
  - stronger result
  
  - implies for all constant $k \geq 2$,
    
    $\text{IP}[k] = \text{AM}[O(k)] = \text{AM}[2]$

- So, $\text{GNI} \in \text{IP}[2] = \text{AM}$

**Back to Graph Isomorphism**

- The payoff:
  
  - not known if GI is $\text{NP}$-complete.
  
  - previous Theorems:
    
    if GI is $\text{NP}$-complete then $\text{PH} = \text{AM}$

  - unlikely!

  - Proof: GI $\text{NP}$-complete $\Rightarrow$ GI $\text{coNP}$-complete
    
    $\Rightarrow \text{coNP} \subseteq \text{AM} \Rightarrow \text{PH} = \text{AM}$