# Seminar on Zero Knowledge

## Interactive Proof Systems and Polynomial Space

Andreas Maletti

May 3, 2019

# Interactive Proof Systems

**Definition (interactive proof system)**

Pair $(A, B)$

- mapping $A\colon \bigcup_{i\in\mathbb{N}}(\{0,1\}^*)^{1+2i} \to \{0,1\}^*$ (Alice)

# Interactive Proof Systems

**Definition (interactive proof system)**

Pair $(A, B)$

- mapping $A\colon \bigcup_{i\in\mathbb{N}}(\{0,1\}^*)^{1+2i} \to \{0,1\}^*$ (Alice)
- randomized polynomial-time algorithm $B$ (of same type) (Bob)

# Interactive Proof Systems

**Definition (interactive proof system)**

Pair $(A, B)$

- mapping $A\colon \bigcup_{i \in \mathbb{N}} (\{0,1\}^*)^{1+2i} \to \{0,1\}^*$ (Alice)
- randomized polynomial-time algorithm $B$ (of same type) (Bob)

Intuition:

- input $w \in \{0,1\}^*$
- polynomial number of rounds (in $n = |w|$)

# Interactive Proof Systems

**Definition (interactive proof system)**

Pair $(A, B)$

- mapping $A\colon \bigcup_{i \in \mathbb{N}} (\{0, 1\}^*)^{1+2i} \to \{0, 1\}^*$ (Alice)
- randomized polynomial-time algorithm $B$ (of same type) (Bob)

Intuition:

- input $w \in \{0, 1\}^*$
- polynomial number of rounds (in $n = |w|$)
- round $i$:
  1. Alice sents $a_i = A(w, a_1, b_1, \dots, a_{i-1}, b_{i-1})$ to Bob

# Interactive Proof Systems

**Definition (interactive proof system)**

Pair $(A, B)$

- mapping $A\colon \bigcup_{i\in\mathbb{N}}(\{0,1\}^*)^{1+2i} \to \{0,1\}^*$ (Alice)
- randomized polynomial-time algorithm $B$ (of same type) (Bob)

Intuition:

- input $w \in \{0,1\}^*$
- polynomial number of rounds (in $n = |w|$)
- round $i$:
  1. Alice sents $a_i = A(w, a_1, b_1, \ldots, a_{i-1}, b_{i-1})$ to Bob
  2. Bob replies with $b_i = B(w, a_1, b_1, \ldots, a_{i-1}, b_{i-1}, a_i)$

# Interactive Proof Systems

**Definition (interactive proof system)**

Pair $(A, B)$

- mapping $A\colon \bigcup_{i \in \mathbb{N}} (\{0,1\}^*)^{1+2i} \to \{0,1\}^*$ (Alice)
- randomized polynomial-time algorithm $B$ (of same type) (Bob)

Intuition:

- input $w \in \{0,1\}^*$
- polynomial number of rounds (in $n = |w|$)
- round $i$:
  1. Alice sents $a_i = A(w, a_1, b_1, \dots, a_{i-1}, b_{i-1})$ to Bob
  2. Bob replies with $b_i = B(w, a_1, b_1, \dots, a_{i-1}, b_{i-1}, a_i)$
- final round: Bob accepts or rejects $w$

Message lengths:

- polynomial length for $b_i$ (since Bob runs in polynomial time)

Message lengths:

- polynomial length for $b_i$ (since Bob runs in polynomial time)
- unbounded $a_i$ but Bob $B$ reads only polynomial prefix

Message lengths:

- polynomial length for $b_i$ (since Bob runs in polynomial time)
- unbounded $a_i$ but Bob $B$ reads only polynomial prefix
- wlog. $|a_i| \leq p(|w|)$ and $|b_i| \leq p(|w|)$ for some polynomial $p$

# Interactive Proof Systems

**Definition (generated language)**

Interactive proof system $(A, B)$ generates language $L \subseteq \{0,1\}^*$ if and only if for all $w \in \{0,1\}^*$:

- Bob rejects $w \in L$ with negligible probability
  (i.e., Bob accepts with probability at least $1 - 2^{-|w|}$)

### Definition (generated language)

Interactive proof system $(A, B)$ generates language $L \subseteq \{0,1\}^*$ if and only if for all $w \in \{0,1\}^*$:

- Bob rejects $w \in L$ with negligible probability
  (i.e., Bob accepts with probability at least $1 - 2^{-|w|}$)
- Bob accepts $w \notin L$ with negligible probability
  in <u>every</u> interactive proof system $(A', B)$

# Interactive Proof Systems

**Definition (generated language)**

Interactive proof system $(A, B)$ generates language $L \subseteq \{0,1\}^*$ if and only if for all $w \in \{0,1\}^*$:

- Bob rejects $w \in L$ with negligible probability (i.e., Bob accepts with probability at least $1 - 2^{-|w|}$)
- Bob accepts $w \notin L$ with negligible probability in <u>every</u> interactive proof system $(A', B)$

$$\mathbf{IP} = \{L \subseteq \{0,1\}^* \mid \exists \text{ interactive proof system generating } L\}$$

Notes:

- introduced by Goldwasser, Micali, and Rackoff in 1985
- Alice is prover and computationally unlimited
- Bob is verifier and restricted to (deterministic) polynomial time

Intuition:

- Completeness:

  If $w \in L$ then designed prover $A$ convinces verifier $B$ almost certainly

# Interactive Proof Systems

Intuition:

- Completeness:

  If $w \in L$ then designed prover $A$ convinces verifier $B$ almost certainly

- Correctness:

  If $w \notin L$ then any prover convinces $B$ only with negligible probability

**Theorem**

$\mathbf{IP} \subseteq \mathbf{PSPACE}$

Proof (1/5)

- $L \in \mathbf{IP}$ and $(A, B)$ interactive proof system generating $L$
- $w \in \{0,1\}^*$ of length $n$ and polynomial $p(x)$ limiting runtime of $B$
- polynomial $q(x)$ limiting number of rounds

**Theorem**

**IP** $\subseteq$ **PSPACE**

Proof (1/5)

- $L \in \mathbf{IP}$ and $(A, B)$ interactive proof system generating $L$
- $w \in \{0,1\}^*$ of length $n$ and polynomial $p(x)$ limiting runtime of $B$
- polynomial $q(x)$ limiting number of rounds
- wlog. length of $a_1, \ldots, a_{q(n)}$ and $b_1, \ldots, b_{q(n)}$ is $p(n)$
- potential Alice

$$A' : \left( \bigcup_{i=0}^{q(n)-1} \{0,1\}^{n+i\cdot 2p(n)} \right) \to \{0,1\}^{p(n)}$$

Proof (2/5)

- approach: construct optimal Alice in PSPACE
- fix Alice $A'$ and random bit sequence $Z \in \{0,1\}^{q(n)p(n)}$ used by Bob
- protocol $P(A', Z) = a_1 b_1 \cdots a_{q(n)} b_{q(n)}$ completely determined

Proof (2/5)

- approach: construct optimal Alice in PSPACE
- fix Alice $A'$ and random bit sequence $Z \in \{0,1\}^{q(n)p(n)}$ used by Bob
- protocol $P(A', Z) = a_1 b_1 \cdots a_{q(n)} b_{q(n)}$ completely determined
- $a_1 b_1 \cdots a_i$ and $a_1 b_1 \cdots a_i b_i$ protocol prefixes for all $0 \leq i \leq q(n)$
- $P \leq P(A', Z)$ if $P$ protocol prefix of $P(A', Z)$

Proof (2/5)

- approach: construct optimal Alice in PSPACE
- fix Alice $A'$ and random bit sequence $Z \in \{0,1\}^{q(n)p(n)}$ used by Bob
- protocol $P(A', Z) = a_1 b_1 \cdots a_{q(n)} b_{q(n)}$ completely determined
- $a_1 b_1 \cdots a_i$ and $a_1 b_1 \cdots a_i b_i$ protocol prefixes for all $0 \leq i \leq q(n)$
- $P \leq P(A', Z)$ if $P$ protocol prefix of $P(A', Z)$

$$f(A', P) = \left| \left\{ Z \in \{0,1\}^{q(n)p(n)} \mid P \leq P(A', Z) \text{ and } P(A', Z) \text{ accepting} \right\} \right|$$

$$f(P) = \max \left\{ f(A'', P) \mid A'' \text{ potential Alice} \right\}$$

- $\bar{A}$ optimal for $P$ if $f(\bar{A}, P) = f(P)$

Proof (2/5)

- approach: construct optimal Alice in PSPACE
- fix Alice $A'$ and random bit sequence $Z \in \{0,1\}^{q(n)p(n)}$ used by Bob
- protocol $P(A', Z) = a_1 b_1 \cdots a_{q(n)} b_{q(n)}$ completely determined
- $a_1 b_1 \cdots a_i$ and $a_1 b_1 \cdots a_i b_i$ protocol prefixes for all $0 \leq i \leq q(n)$
- $P \leq P(A', Z)$ if $P$ protocol prefix of $P(A', Z)$

$$f(A', P) = \left| \left\{ Z \in \{0,1\}^{q(n)p(n)} \mid P \leq P(A', Z) \text{ and } P(A', Z) \text{ accepting} \right\} \right|$$

$$f(P) = \max \left\{ f(A'', P) \mid A'' \text{ potential Alice} \right\}$$

- $\bar{A}$ optimal for $P$ if $f(\bar{A}, P) = f(P)$
- $w$ accepted with probability $\rho(w) \leq \frac{f(\varepsilon)}{2^{q(n)p(n)}}$

Proof (3/5)

- $\frac{f(\varepsilon)}{2^{q(n)p(n)}} \geq \rho(w) \geq 1 - 2^{-n}$ and $f(\varepsilon) \geq (1 - 2^{-n}) \cdot 2^{q(n)p(n)}$ for all $w \in L$
- otherwise $f(\varepsilon) \leq 2^{-n} \cdot 2^{q(n)p(n)}$
- compute $f(\varepsilon)$ recursively in polynomial space

Proof (3/5)

- $\frac{f(\varepsilon)}{2^{q(n)p(n)}} \geq \rho(w) \geq 1 - 2^{-n}$ and $f(\varepsilon) \geq (1 - 2^{-n}) \cdot 2^{q(n)p(n)}$ for all $w \in L$
- otherwise $f(\varepsilon) \leq 2^{-n} \cdot 2^{q(n)p(n)}$
- compute $f(\varepsilon)$ recursively in polynomial space

1. complete protocol $P = a_1 b_1 \cdots a_{q(n)} b_{q(n)}$
   - if Bob rejects, then $f(P) = 0$.
   - otherwise

$$f(P) = \left| \{ Z \in \{0,1\}^{q(n)p(n)} \mid Z \text{ permits } P \} \right|$$

$Z$ permits $P$ if bit sequence $Z$ yields the responses $b_1, \ldots, b_{q(n)}$ from $B$

<u>Proof</u> (3/5)

- $\frac{f(\varepsilon)}{2^{q(n)p(n)}} \geq \rho(w) \geq 1 - 2^{-n}$ and $f(\varepsilon) \geq (1 - 2^{-n}) \cdot 2^{q(n)p(n)}$ for all $w \in L$
- otherwise $f(\varepsilon) \leq 2^{-n} \cdot 2^{q(n)p(n)}$
- compute $f(\varepsilon)$ recursively in polynomial space

1. complete protocol $P = a_1 b_1 \cdots a_{q(n)} b_{q(n)}$
   - if Bob rejects, then $f(P) = 0$.
   - otherwise

$$f(P) = \left| \left\{ Z \in \{0,1\}^{q(n)p(n)} \mid Z \text{ permits } P \right\} \right|$$

     $Z$ permits $P$ if bit sequence $Z$ yields the responses $b_1, \ldots, b_{q(n)}$ from $B$
   - can be done in polynomial space

Proof (4/5)

2. incomplete protocol $P = a_1 b_1 \cdots a_{i-1} b_{i-1} a_i$ with final message from Alice $1 \leq i \leq q(n)$
   - $f(Pb_i)$ with $b_i \in \{0,1\}^{p(n)}$ known and

Proof (4/5)

2. incomplete protocol $P = a_1 b_1 \cdots a_{i-1} b_{i-1} a_i$ with final message from Alice $\qquad 1 \leq i \leq q(n)$
   - $f(Pb_i)$ with $b_i \in \{0,1\}^{p(n)}$ known and

$$
\begin{aligned}
f(P) &= \max\left\{ f(A'', P) \mid \text{potential Alice } A'' \right\} \\
&= \max\left\{ \sum_{b_i \in \{0,1\}^{p(n)}} f(A'', Pb_i) \mid \text{potential Alice } A'' \right\} \\
&= \sum_{b_i \in \{0,1\}^{p(n)}} \max\{ f(A''_{b_i}, Pb_i) \mid \text{potential Alice } A''_{b_i} \} \\
&= \sum_{b_i \in \{0,1\}^{p(n)}} f(Pb_i)
\end{aligned}
$$

   - second-to-last equality follows because $Z$ determines protocol, so a single $Z$ cannot permit $Pb_i$ and $Pb'_i$ with $b_i \neq b'_i$

Proof (4/5)

2. incomplete protocol $P = a_1 b_1 \cdots a_{i-1} b_{i-1} a_i$
   with final message from Alice $1 \leq i \leq q(n)$
   - $f(Pb_i)$ with $b_i \in \{0,1\}^{p(n)}$ known and

$$\begin{aligned} f(P) &= \max \left\{ f(A'', P) \mid \text{potential Alice } A'' \right\} \\ &= \max\left\{ \sum_{b_i \in \{0,1\}^{p(n)}} f(A'', Pb_i) \mid \text{potential Alice } A'' \right\} \\ &= \sum_{b_i \in \{0,1\}^{p(n)}} \max\{ f(A''_{b_i}, Pb_i) \mid \text{potential Alice } A''_{b_i} \} \\ &= \sum_{b_i \in \{0,1\}^{p(n)}} f(Pb_i) \end{aligned}$$

   - second-to-last equality follows because $Z$ determines protocol, so a single $Z$ cannot permit $Pb_i$ and $Pb'_i$ with $b_i \neq b'_i$
   - clearly also polynomial space

Proof (5/5)

3. incomplete protocol $P = a_1 b_1 \cdots a_{i-1} b_{i-1}$
   with final message from Bob $1 \leq i \leq q(n)$
   - $f(Pa_i)$ with $a_i \in \{0,1\}^{p(n)}$ known and Alice can select the response

$$f(P) = \max \left\{ f(Pa_i) \mid a_i \in \{0,1\}^{p(n)} \right\}$$

   - clearly also polynomial space

Proof (5/5)

3. incomplete protocol $P = a_1 b_1 \cdots a_{i-1} b_{i-1}$ with final message from Bob $\quad 1 \leq i \leq q(n)$
   - $f(Pa_i)$ with $a_i \in \{0,1\}^{p(n)}$ known and Alice can select the response

$$f(P) = \max \left\{ f(Pa_i) \mid a_i \in \{0,1\}^{p(n)} \right\}$$

   - clearly also polynomial space

Space requirements:

- recursion depth $2q(n)$
- protocol prefix $P$, messages $a_i$ and $b_i$
- currently best value of $f$ and partial sum both limited by $2^{q(n)p(n)}$ $\square$

**Lemma**

**IP** closed under polynomial-time reductions

$$L \in \mathbf{IP} \qquad \text{for all} \qquad L \preceq_{\mathbf{P}} L' \quad \text{and} \quad L' \in \mathbf{IP}$$

<u>Proof</u>

- $f$ polynomial-time reduction of $L$ to $L'$
- $(A, B)$ interactive proof system generating $L'$ and input $w$

### Lemma

**IP** closed under polynomial-time reductions

$$L \in \mathbf{IP} \qquad \text{for all} \qquad L \preceq_{\mathbf{P}} L' \quad \text{and} \quad L' \in \mathbf{IP}$$

<u>Proof</u>

- $f$ polynomial-time reduction of $L$ to $L'$
- $(A, B)$ interactive proof system generating $L'$ and input $w$
- Alice $A'$ and Bob $B'$ compute $f(w)$ and then simulate $A$ and $B$
- yields answer for $f(w) \overset{?}{\in} L'$ that is correct for $w \overset{?}{\in} L$ $\square$

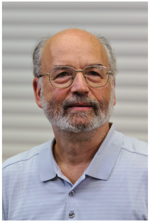
# Interactive Proof Systems

**Theorem (Shamir 1990)**

$\mathbf{IP} = \mathbf{PSPACE}$

**Adi Shamir (* 1952)**

- isra. computer scientist
- professor at Weizmann institute and ENS Paris
- Turing laureate 2002 and 'S' in RSA



© Erik Tews

Proof (1/7)

- we know **IP** $\subseteq$ **PSPACE**, so only **PSPACE** $\subseteq$ **IP** remains
- since **IP** is closed under polynomial-time reductions and QBF is **PSPACE**-complete, we just show QBF $\in$ **IP**

Proof (1/7)

- we know **IP** $\subseteq$ **PSPACE**, so only **PSPACE** $\subseteq$ **IP** remains
- since **IP** is closed under polynomial-time reductions and QBF is **PSPACE**-complete, we just show QBF $\in$ **IP**
- let $F$ closed quantified formula over $\wedge$, $\vee$, $\forall$, $\exists$ and the literals

Proof (1/7)

- we know **IP** $\subseteq$ **PSPACE**, so only **PSPACE** $\subseteq$ **IP** remains
- since **IP** is closed under polynomial-time reductions and QBF is **PSPACE**-complete, we just show QBF $\in$ **IP**
- let $F$ closed quantified formula over $\wedge$, $\vee$, $\forall$, $\exists$ and the literals
- replace $F$ by arithmetic expression $a(F)$
  - $a(x) = x$ and $a(\neg x) = 1 - x$ for all variables $x$

Proof (1/7)

- we know **IP** $\subseteq$ **PSPACE**, so only **PSPACE** $\subseteq$ **IP** remains
- since **IP** is closed under polynomial-time reductions and QBF is **PSPACE**-complete, we just show QBF $\in$ **IP**
- let $F$ closed quantified formula over $\wedge$, $\vee$, $\forall$, $\exists$ and the literals
- replace $F$ by arithmetic expression $a(F)$
  - $a(x) = x$ and $a(\neg x) = 1 - x$ for all variables $x$
  - $a(F_1 \vee F_2) = a(F_1) + a(F_2)$ and $a(F_1 \wedge F_2) = a(F_1) \cdot a(F_2)$ for all $F_1$ and $F_2$

Proof (1/7)

- we know **IP** $\subseteq$ **PSPACE**, so only **PSPACE** $\subseteq$ **IP** remains
- since **IP** is closed under polynomial-time reductions and QBF is **PSPACE**-complete, we just show QBF $\in$ **IP**
- let $F$ closed quantified formula over $\wedge$, $\vee$, $\forall$, $\exists$ and the literals
- replace $F$ by arithmetic expression $a(F)$
  - $a(x) = x$ and $a(\neg x) = 1 - x$ for all variables $x$
  - $a(F_1 \vee F_2) = a(F_1) + a(F_2)$ and $a(F_1 \wedge F_2) = a(F_1) \cdot a(F_2)$ for all $F_1$ and $F_2$
  - $a(\exists x F_1) = \sum_{x \in \{0,1\}} a(F_1)$ and $a(\forall x F_1) = \prod_{x \in \{0,1\}} a(F_1)$ for all $F_1$

Proof (1/7)

- we know $\mathbf{IP} \subseteq \mathbf{PSPACE}$, so only $\mathbf{PSPACE} \subseteq \mathbf{IP}$ remains
- since **IP** is closed under polynomial-time reductions and QBF is **PSPACE**-complete, we just show $\text{QBF} \in \mathbf{IP}$
- let $F$ closed quantified formula over $\wedge, \vee, \forall, \exists$ and the literals
- replace $F$ by arithmetic expression $a(F)$
  - $a(x) = x$ and $a(\neg x) = 1 - x$ for all variables $x$
  - $a(F_1 \vee F_2) = a(F_1) + a(F_2)$ and $a(F_1 \wedge F_2) = a(F_1) \cdot a(F_2)$ for all $F_1$ and $F_2$
  - $a(\exists x F_1) = \sum_{x \in \{0,1\}} a(F_1)$ and $a(\forall x F_1) = \prod_{x \in \{0,1\}} a(F_1)$ for all $F_1$
- obviously $a(F) \geq 0$
- $F \in \text{QBF}$ if and only if $a(F) > 0$

Example: $F = \forall x \exists y \Big( (x \vee \neg y) \wedge \exists z(\neg x \wedge z) \Big)$.

$$\begin{aligned}
a(F) &= \prod_{x \in \{0,1\}} \left( \sum_{y \in \{0,1\}} \Big( (x + (1-y)) \cdot \sum_{z \in \{0,1\}} ((1-x) \cdot z) \Big) \right) \\
&= \prod_{x \in \{0,1\}} \left( \sum_{y \in \{0,1\}} \Big( (x + (1-y)) \cdot (1-x) \Big) \right) \\
&= \prod_{x \in \{0,1\}} \Big( (1-x^2) + (x - x^2) \Big) \\
&= 0
\end{aligned}$$

so the formula is “wrong”

Example: For the negation $\neg F = \exists x \forall y \Big( (\neg x \wedge y) \vee \forall z (x \vee \neg z) \Big)$

$$\begin{aligned}
a(\neg F) &= \sum_{x \in \{0,1\}} \left( \prod_{y \in \{0,1\}} \Big( (1-x) \cdot y + \prod_{z \in \{0,1\}} \big( x + (1-z) \big) \Big) \right) \\
&= \sum_{x \in \{0,1\}} \left( \prod_{y \in \{0,1\}} \Big( (1-x) \cdot y + (x^2 + x) \Big) \right) \\
&= \sum_{x \in \{0,1\}} (x^2 + x) \cdot (1 + x^2) \\
&= 4
\end{aligned}$$

so the negated formula is “true”

How large can $a(F)$ be?

How large can $a(F)$ be? For formula $F$ the length $|F|$ is

- $|0| = |1| = |x| = |\neg x| = 1$
- $|F \vee G| = |F \wedge G| = |F| + |G|$
- $|\exists x F| = |\forall x F| = 1 + |F|$

# Interactive Proof Systems

How large can $a(F)$ be? For formula $F$ the length $|F|$ is

- $|0| = |1| = |x| = |\neg x| = 1$
- $|F \vee G| = |F \wedge G| = |F| + |G|$
- $|\exists x F| = |\forall x F| = 1 + |F|$

**Lemma**

$a(F) \leq 2^{2^{|F|}}$

Proof

- replace each occurrence of $\exists x G$ by $G[x \mapsto 0] \vee G[x \mapsto 1]$ and each $\forall x G$ by $G[x \mapsto 0] \wedge G[x \mapsto 1]$
- prove $|F'| \leq 2^{|F|}$ for obtained formula $F'$ by induction

How large can $a(F)$ be? For formula $F$ the length $|F|$ is

- $|0| = |1| = |x| = |\neg x| = 1$
- $|F \vee G| = |F \wedge G| = |F| + |G|$
- $|\exists x F| = |\forall x F| = 1 + |F|$

**Lemma**

$a(F) \leq 2^{2^{|F|}}$

Proof

- replace each occurrence of $\exists x G$ by $G[x \mapsto 0] \vee G[x \mapsto 1]$ and each $\forall x G$ by $G[x \mapsto 0] \wedge G[x \mapsto 1]$
- prove $|F'| \leq 2^{|F|}$ for obtained formula $F'$ by induction
- prove $a(F') \leq 2^{|F'|}$ by induction $\square$

Example: $F = \forall x_1 \cdots \forall x_k \exists y \exists z (y \vee z)$.

$$\begin{aligned}
a(F) &= \prod_{x_1 \in \{0,1\}} \cdots \prod_{x_k \in \{0,1\}} \left( \sum_{y \in \{0,1\}} \sum_{z \in \{0,1\}} (y+z) \right) \\
&= \prod_{x_1 \in \{0,1\}} \cdots \prod_{x_k \in \{0,1\}} \left( \sum_{y \in \{0,1\}} (2y+1) \right) \\
&= \prod_{x_1 \in \{0,1\}} \cdots \prod_{x_k \in \{0,1\}} 4 \\
&= 4^{2^k}
\end{aligned}$$

Notes:

- numbers of size $2^{2^{|F|}}$ require $2^{|F|}$ bits
- cannot be exchanged in protocol round

# Interactive Proof Systems

Notes:

- numbers of size $2^{2^{|F|}}$ require $2^{|F|}$ bits
- cannot be exchanged in protocol round
- compute modulo prime

**Lemma [Dietzfelbinger 2004]**

For $n \geq 5$ interval $[2^n, 2^{2n}]$ contains at least $2^n$ primes

Proof (2/7)

- $n = |F|$ and $p_1, \ldots, p_k$ primes between $2^n$ and $2^{2n}$

Proof (2/7)

- $n = |F|$ and $p_1, \ldots, p_k$ primes between $2^n$ and $2^{2n}$
- $m = \prod_{i=1}^{k} p_i \geq (2^n)^{(2^n)} = 2^{n \cdot 2^n} > 2^{2^n} \geq a(F)$

Proof (2/7)

- $n = |F|$ and $p_1, \ldots, p_k$ primes between $2^n$ and $2^{2n}$
- $m = \prod_{i=1}^{k} p_i \geq (2^n)^{(2^n)} = 2^{n \cdot 2^n} > 2^{2^n} \geq a(F)$

$$F \notin \mathrm{QBF} \iff a(F) \equiv 0 \bmod m \qquad \text{since } a(F) = 0$$
$$F \in \mathrm{QBF} \iff \exists 1 \leq i \leq k \colon a(F) \not\equiv 0 \bmod p_i$$

because $m = \prod_{i=1}^{k} p_i > a(F)$, so not all $p_i$ can divide $a(F)$

Proof (3/7)

- for $F \in \mathrm{QBF}$ Alice computes smallest prime $p_i \geq 2^n$ with $a(F) \not\equiv 0 \bmod p_i$
- sends $p_i \leq 2^{2n}$ to Bob
- all other computations now modulo $p_i$

Proof (3/7)

- for $F \in \text{QBF}$ Alice computes smallest prime $p_i \geq 2^n$ with $a(F) \not\equiv 0 \bmod p_i$
- sends $p_i \leq 2^{2n}$ to Bob
- all other computations now modulo $p_i$
- wlog. $F = QxF'$ with $Q \in \{\exists, \forall\}$
- polynomial $a(F')$ obtained from $a(F)$ by removing first product $\prod$ or sum $\sum$

Proof (3/7)

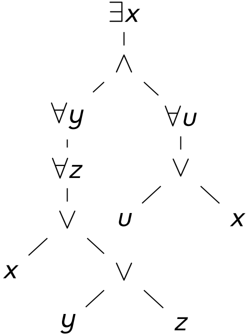
- for $F \in \text{QBF}$ Alice computes smallest prime $p_i \geq 2^n$ with $a(F) \not\equiv 0 \bmod p_i$
- sends $p_i \leq 2^{2n}$ to Bob
- all other computations now modulo $p_i$
- wlog. $F = QxF'$ with $Q \in \{\exists, \forall\}$
- polynomial $a(F')$ obtained from $a(F)$ by removing first product $\prod$ or sum $\sum$
- $\deg\big(a(F')\big)$ can be exponential in $n$

## Definition (simple formula)

Formula is simple if at most one additional $\forall$-quantifier occurs between quantification $Qx$ with $Q \in \{\exists, \forall\}$ and each occurrence of variable $x$

**Definition (simple formula)**

Formula is simple if at most one additional $\forall$-quantifier occurs between quantification $Qx$ with $Q \in \{\exists, \forall\}$ and each occurrence of variable $x$




$\exists x \Big(\forall y \forall z \big(x \vee (y \vee z)\big)\Big) \wedge \Big(\forall u \big(u \vee x\big)\Big)$ <u>not</u> simple

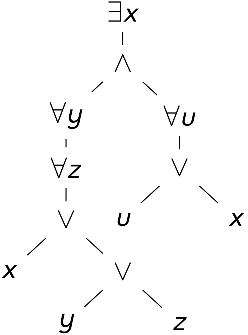
**Lemma**

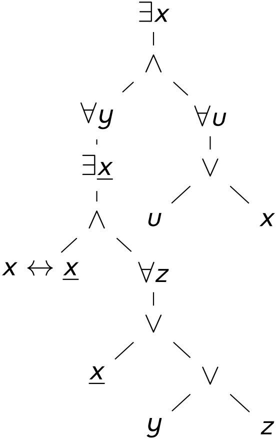
Each formula can be transformed into equivalent simple formula in polynomial time

Proof

- replace each subformula $\forall y G(x_1, \ldots, x_k, y)$ with free variables $y, x_1, \ldots, x_k$ by

$$\forall y \exists y_1 \cdots \exists y_k \Big( \bigwedge_{i=1}^{k} x_i \leftrightarrow y_i \wedge G(y_1, \ldots, y_k, y) \Big) \qquad \square$$


∃x
∧
∀y
∀u
∀z
∨
∨
u
x
x
∨
y
z


∃x
∧
∀y
∀u
∃x
∨
∧
u
x
x ↔ x
∀z
∨
x
∨
y
z

**Lemma**

$\deg\big(a(G')\big) \leq 2|G|$ for simple formula $G = QxG'$ with $Q \in \{\exists, \forall\}$

Proof

- replace in $G'$ each subformula $\forall yH$ in which $x$ occurs freely by $H[y \mapsto 0] \wedge H[y \mapsto 1]$

**Lemma**

$\deg\big(a(G')\big) \le 2|G|$ for simple formula $G = QxG'$ with $Q \in \{\exists, \forall\}$

Proof

- replace in $G'$ each subformula $\forall yH$ in which $x$ occurs freely by $H[y \mapsto 0] \wedge H[y \mapsto 1]$
- doubles length of formula since those subformulas are not nested

## Lemma

$\deg\big(a(G')\big) \leq 2|G|$ for simple formula $G = QxG'$ with $Q \in \{\exists, \forall\}$

Proof

- replace in $G'$ each subformula $\forall yH$ in which $x$ occurs freely by $H[y \mapsto 0] \wedge H[y \mapsto 1]$
- doubles length of formula since those subformulas are not nested
- show $\deg\big(a(G'')\big) \leq |G''|$ for obtained formula $G''$
  1. $\deg\big(a(x)\big) = \deg\big(a(0)\big) = \deg\big(a(1)\big) = 1$

**Lemma**

$\deg\big(a(G')\big) \leq 2|G|$ for simple formula $G = QxG'$ with $Q \in \{\exists, \forall\}$

Proof

- replace in $G'$ each subformula $\forall yH$ in which $x$ occurs freely by $H[y \mapsto 0] \wedge H[y \mapsto 1]$
- doubles length of formula since those subformulas are not nested
- show $\deg\big(a(G'')\big) \leq |G''|$ for obtained formula $G''$
  1. $\deg\big(a(x)\big) = \deg\big(a(0)\big) = \deg\big(a(1)\big) = 1$
  2. $\deg\big(a(G_1 \vee G_2)\big) \leq \max\{|G_1|, |G_2|\} < |G''|$

**Lemma**

$\deg\big(a(G')\big) \le 2|G|$ for simple formula $G = QxG'$ with $Q \in \{\exists, \forall\}$

Proof

- replace in $G'$ each subformula $\forall y H$ in which $x$ occurs freely by $H[y \mapsto 0] \wedge H[y \mapsto 1]$
- doubles length of formula since those subformulas are not nested
- show $\deg\big(a(G'')\big) \le |G''|$ for obtained formula $G''$
  1. $\deg\big(a(x)\big) = \deg\big(a(0)\big) = \deg\big(a(1)\big) = 1$
  2. $\deg\big(a(G_1 \vee G_2)\big) \le \max\{|G_1|, |G_2|\} < |G''|$
  3. $\deg\big(a(G_1 \wedge G_2)\big) \le |G_1| + |G_2| = |G''|$

**Lemma**

$\deg\big(a(G')\big) \leq 2|G|$ for simple formula $G = QxG'$ with $Q \in \{\exists, \forall\}$

Proof

- replace in $G'$ each subformula $\forall yH$ in which $x$ occurs freely by $H[y \mapsto 0] \wedge H[y \mapsto 1]$
- doubles length of formula since those subformulas are not nested
- show $\deg\big(a(G'')\big) \leq |G''|$ for obtained formula $G''$
  1. $\deg\big(a(x)\big) = \deg\big(a(0)\big) = \deg\big(a(1)\big) = 1$
  2. $\deg\big(a(G_1 \vee G_2)\big) \leq \max\{|G_1|, |G_2|\} < |G''|$
  3. $\deg\big(a(G_1 \wedge G_2)\big) \leq |G_1| + |G_2| = |G''|$
  4. $\deg\big(a(\exists yG_1)\big) \leq \deg(a(G_1)) < |G''|$ for $y \neq x$

**Lemma**

$\deg\big(a(G')\big) \le 2|G|$ for simple formula $G = QxG'$ with $Q \in \{\exists, \forall\}$

Proof

- replace in $G'$ each subformula $\forall yH$ in which $x$ occurs freely by $H[y \mapsto 0] \wedge H[y \mapsto 1]$
- doubles length of formula since those subformulas are not nested
- show $\deg\big(a(G'')\big) \le |G''|$ for obtained formula $G''$
  1. $\deg\big(a(x)\big) = \deg\big(a(0)\big) = \deg\big(a(1)\big) = 1$
  2. $\deg\big(a(G_1 \vee G_2)\big) \le \max\{|G_1|, |G_2|\} < |G''|$
  3. $\deg\big(a(G_1 \wedge G_2)\big) \le |G_1| + |G_2| = |G''|$
  4. $\deg\big(a(\exists y G_1)\big) \le \deg(a(G_1)) < |G''|$ for $y \ne x$
  5. $\deg\big(a(\forall y G_1)\big) = 0 < |G''|$ since $x$ does not occur in $G_1$ $\square$

Proof (4/7)

- play $\ell$ rounds with $\ell \leq n$ the number of quantifiers in $F$
- let $F = F_1 = Q_1 x_1 G_1$ with $\rho_1(x_1) = a(G_1) \bmod p_i$
  a polynomial (in $x_1$) of degree at most $2n$

Proof (4/7)

- play $\ell$ rounds with $\ell \leq n$ the number of quantifiers in $F$
- let $F = F_1 = Q_1 x_1 G_1$ with $\rho_1(x_1) = a(G_1) \bmod p_i$
  a polynomial (in $x_1$) of degree at most $2n$
- start rounds
  1. Alice sends $a_1 = a(F_1) \bmod p_i$

Proof (4/7)

- play $\ell$ rounds with $\ell \leq n$ the number of quantifiers in $F$
- let $F = F_1 = Q_1 x_1 G_1$ with $\rho_1(x_1) = a(G_1) \bmod p_i$
  a polynomial (in $x_1$) of degree at most $2n$
- start rounds
  1. Alice sends $a_1 = a(F_1) \bmod p_i$
  2. Bob rejects $F$ if $a_1 = 0$; otherwise demands proof for $a_1$

Proof (4/7)

- play $\ell$ rounds with $\ell \leq n$ the number of quantifiers in $F$
- let $F = F_1 = Q_1 x_1 G_1$ with $\rho_1(x_1) = a(G_1) \bmod p_i$
  a polynomial (in $x_1$) of degree at most $2n$
- start rounds
  1. Alice sends $a_1 = a(F_1) \bmod p_i$
  2. Bob rejects $F$ if $a_1 = 0$; otherwise demands proof for $a_1$
  3. Alice sends polynomial $\rho_1(x_1)$

Proof (4/7)

- play $\ell$ rounds with $\ell \leq n$ the number of quantifiers in $F$
- let $F = F_1 = Q_1 x_1 G_1$ with $\rho_1(x_1) = a(G_1) \bmod p_i$
  a polynomial (in $x_1$) of degree at most $2n$
- start rounds
  1. Alice sends $a_1 = a(F_1) \bmod p_i$
  2. Bob rejects $F$ if $a_1 = 0$; otherwise demands proof for $a_1$
  3. Alice sends polynomial $\rho_1(x_1)$
  4. For $Q_1 = \exists$ Bob checks $a_1 \equiv \rho_1(0) + \rho_1(1) \bmod p_i$

Proof (4/7)

- play $\ell$ rounds with $\ell \leq n$ the number of quantifiers in $F$
- let $F = F_1 = Q_1 x_1 G_1$ with $\rho_1(x_1) = a(G_1) \bmod p_i$
  a polynomial (in $x_1$) of degree at most $2n$
- start rounds
  1. Alice sends $a_1 = a(F_1) \bmod p_i$
  2. Bob rejects $F$ if $a_1 = 0$; otherwise demands proof for $a_1$
  3. Alice sends polynomial $\rho_1(x_1)$
  4. For $Q_1 = \exists$ Bob checks $a_1 \equiv \rho_1(0) + \rho_1(1) \bmod p_i$
  5. For $Q_1 = \forall$ Bob checks $a_1 \equiv \rho_1(0) \cdot \rho_1(1) \bmod p_i$
  6. Bob randomly selects $0 \leq r_1 < p_i$, shares it with Alice and computes $\rho_1(r_1) \bmod p_i$

Proof (4/7)

- play $\ell$ rounds with $\ell \leq n$ the number of quantifiers in $F$
- let $F = F_1 = Q_1 x_1 G_1$ with $\rho_1(x_1) = a(G_1) \bmod p_i$
  a polynomial (in $x_1$) of degree at most $2n$
- start rounds
  1. Alice sends $a_1 = a(F_1) \bmod p_i$
  2. Bob rejects $F$ if $a_1 = 0$; otherwise demands proof for $a_1$
  3. Alice sends polynomial $\rho_1(x_1)$
  4. For $Q_1 = \exists$ Bob checks $a_1 \equiv \rho_1(0) + \rho_1(1) \bmod p_i$
  5. For $Q_1 = \forall$ Bob checks $a_1 \equiv \rho_1(0) \cdot \rho_1(1) \bmod p_i$
  6. Bob randomly selects $0 \leq r_1 < p_i$, shares it with Alice and computes $\rho_1(r_1) \bmod p_i$
  7. write $a(G_1)[x_1 \mapsto r_1]$ as $b + c \cdot a(F_2)[x_1 \mapsto r_1]$
     with $F_2$ subformula of $G_1$ starting with first quantifier
  8. Bob computes $0 \leq b, c < p_i$

Proof (4/7)

- play $\ell$ rounds with $\ell \leq n$ the number of quantifiers in $F$
- let $F = F_1 = Q_1 x_1 G_1$ with $\rho_1(x_1) = a(G_1) \bmod p_i$
  a polynomial (in $x_1$) of degree at most $2n$
- start rounds
  1. Alice sends $a_1 = a(F_1) \bmod p_i$
  2. Bob rejects $F$ if $a_1 = 0$; otherwise demands proof for $a_1$
  3. Alice sends polynomial $\rho_1(x_1)$
  4. For $Q_1 = \exists$ Bob checks $a_1 \equiv \rho_1(0) + \rho_1(1) \bmod p_i$
  5. For $Q_1 = \forall$ Bob checks $a_1 \equiv \rho_1(0) \cdot \rho_1(1) \bmod p_i$
  6. Bob randomly selects $0 \leq r_1 < p_i$, shares it with Alice and computes $\rho_1(r_1) \bmod p_i$
  7. write $a(G_1)[x_1 \mapsto r_1]$ as $b + c \cdot a(F_2)[x_1 \mapsto r_1]$
     with $F_2$ subformula of $G_1$ starting with first quantifier
  8. Bob computes $0 \leq b, c < p_i$
  9. Bob accepts if $c = 0$ and $\rho_1(r_1) = b$

Proof (4/7)

- play $\ell$ rounds with $\ell \leq n$ the number of quantifiers in $F$
- let $F = F_1 = Q_1 x_1 G_1$ with $\rho_1(x_1) = a(G_1) \bmod p_i$
  a polynomial (in $x_1$) of degree at most $2n$
- start rounds
  1. Alice sends $a_1 = a(F_1) \bmod p_i$
  2. Bob rejects $F$ if $a_1 = 0$; otherwise demands proof for $a_1$
  3. Alice sends polynomial $\rho_1(x_1)$
  4. For $Q_1 = \exists$ Bob checks $a_1 \equiv \rho_1(0) + \rho_1(1) \bmod p_i$
  5. For $Q_1 = \forall$ Bob checks $a_1 \equiv \rho_1(0) \cdot \rho_1(1) \bmod p_i$
  6. Bob randomly selects $0 \leq r_1 < p_i$, shares it with Alice and computes $\rho_1(r_1) \bmod p_i$
  7. write $a(G_1)[x_1 \mapsto r_1]$ as $b + c \cdot a(F_2)[x_1 \mapsto r_1]$
     with $F_2$ subformula of $G_1$ starting with first quantifier
  8. Bob computes $0 \leq b, c < p_i$
  9. Bob accepts if $c = 0$ and $\rho_1(r_1) = b$
  10. otherwise Bob computes $a_2 = (\rho_1(r_1) - b) \cdot c^{-1} \bmod p_i$

Proof (5/7)

- for correct polynomial $\rho_1(x_1) = a(G_1)$

$$a(F_2)[x_1 \mapsto r_1] = \frac{a(G_1)[x_1 \mapsto r_1] - b}{c} = \frac{\rho_1(r_1) - b}{c} = a_2 \bmod p_i$$

- let $F_2 = Q_2 x_2 G_2$ and $\rho_2(x_2) = a(G_2)[x_1 \mapsto r_1]$
- start round 2
  2. Bob demands proof for $a(F_2)[x_1 \mapsto r_1] = a_2 \bmod p_i$

Proof (5/7)

- for correct polynomial $\rho_1(x_1) = a(G_1)$

$$a(F_2)[x_1 \mapsto r_1] = \frac{a(G_1)[x_1 \mapsto r_1] - b}{c} = \frac{\rho_1(r_1) - b}{c} = a_2 \bmod p_i$$

- let $F_2 = Q_2 x_2 G_2$ and $\rho_2(x_2) = a(G_2)[x_1 \mapsto r_1]$
- start round 2
  2. Bob demands proof for $a(F_2)[x_1 \mapsto r_1] = a_2 \bmod p_i$
  3. Alice sends polynomial $\rho_2(x_2)$

Proof (5/7)

- for correct polynomial $\rho_1(x_1) = a(G_1)$

$$a(F_2)[x_1 \mapsto r_1] = \frac{a(G_1)[x_1 \mapsto r_1] - b}{c} = \frac{\rho_1(r_1) - b}{c} = a_2 \bmod p_i$$

- let $F_2 = Q_2 x_2 G_2$ and $\rho_2(x_2) = a(G_2)[x_1 \mapsto r_1]$
- start round 2
  2. Bob demands proof for $a(F_2)[x_1 \mapsto r_1] = a_2 \bmod p_i$
  3. Alice sends polynomial $\rho_2(x_2)$
  4. For $Q_2 = \exists$ Bob checks $a_2 \equiv \rho_2(0) + \rho_2(1) \bmod p_i$

Proof (5/7)

- for correct polynomial $\rho_1(x_1) = a(G_1)$

$$a(F_2)[x_1 \mapsto r_1] = \frac{a(G_1)[x_1 \mapsto r_1] - b}{c} = \frac{\rho_1(r_1) - b}{c} = a_2 \bmod p_i$$

- let $F_2 = Q_2 x_2 G_2$ and $\rho_2(x_2) = a(G_2)[x_1 \mapsto r_1]$
- start round 2
  2. Bob demands proof for $a(F_2)[x_1 \mapsto r_1] = a_2 \bmod p_i$
  3. Alice sends polynomial $\rho_2(x_2)$
  4. For $Q_2 = \exists$ Bob checks $a_2 \equiv \rho_2(0) + \rho_2(1) \bmod p_i$
  5. For $Q_2 = \forall$ Bob checks $a_2 \equiv \rho_2(0) \cdot \rho_2(1) \bmod p_i$
  6. Bob randomly selects $0 \leq r_2 < p_i$, shares it with Alice and computes $\rho_2(r_2) \bmod p_i$

Proof (5/7)

- for correct polynomial $\rho_1(x_1) = a(G_1)$

$$a(F_2)[x_1 \mapsto r_1] = \frac{a(G_1)[x_1 \mapsto r_1] - b}{c} = \frac{\rho_1(r_1) - b}{c} = a_2 \bmod p_i$$

- let $F_2 = Q_2 x_2 G_2$ and $\rho_2(x_2) = a(G_2)[x_1 \mapsto r_1]$
- start round 2
  2. Bob demands proof for $a(F_2)[x_1 \mapsto r_1] = a_2 \bmod p_i$
  3. Alice sends polynomial $\rho_2(x_2)$
  4. For $Q_2 = \exists$ Bob checks $a_2 \equiv \rho_2(0) + \rho_2(1) \bmod p_i$
  5. For $Q_2 = \forall$ Bob checks $a_2 \equiv \rho_2(0) \cdot \rho_2(1) \bmod p_i$
  6. Bob randomly selects $0 \leq r_2 < p_i$, shares it with Alice and computes $\rho_2(r_2) \bmod p_i$
  7. write $a(G_2)[x_1 \mapsto r_1, x_2 \mapsto r_2]$ as $b + c \cdot a(F_3)[x_1 \mapsto r_1, x_2 \mapsto r_2]$ with $F_3$ subformula of $G_2$ starting with first quantifier
  8. Bob computes $0 \leq b, c < p_i$

Proof (5/7)

- for correct polynomial $\rho_1(x_1) = a(G_1)$

$$a(F_2)[x_1 \mapsto r_1] = \frac{a(G_1)[x_1 \mapsto r_1] - b}{c} = \frac{\rho_1(r_1) - b}{c} = a_2 \bmod p_i$$

- let $F_2 = Q_2 x_2 G_2$ and $\rho_2(x_2) = a(G_2)[x_1 \mapsto r_1]$
- start round 2
  2. Bob demands proof for $a(F_2)[x_1 \mapsto r_1] = a_2 \bmod p_i$
  3. Alice sends polynomial $\rho_2(x_2)$
  4. For $Q_2 = \exists$ Bob checks $a_2 \equiv \rho_2(0) + \rho_2(1) \bmod p_i$
  5. For $Q_2 = \forall$ Bob checks $a_2 \equiv \rho_2(0) \cdot \rho_2(1) \bmod p_i$
  6. Bob randomly selects $0 \leq r_2 < p_i$, shares it with Alice and computes $\rho_2(r_2) \bmod p_i$
  7. write $a(G_2)[x_1 \mapsto r_1, x_2 \mapsto r_2]$ as $b + c \cdot a(F_3)[x_1 \mapsto r_1, x_2 \mapsto r_2]$ with $F_3$ subformula of $G_2$ starting with first quantifier
  8. Bob computes $0 \leq b, c < p_i$
  9. Bob accepts if $c = 0$ and $\rho_2(r_2) = b$

Proof (5/7)

- for correct polynomial $\rho_1(x_1) = a(G_1)$

$$a(F_2)[x_1 \mapsto r_1] = \frac{a(G_1)[x_1 \mapsto r_1] - b}{c} = \frac{\rho_1(r_1) - b}{c} = a_2 \bmod p_i$$

- let $F_2 = Q_2 x_2 G_2$ and $\rho_2(x_2) = a(G_2)[x_1 \mapsto r_1]$
- start round 2
  2. Bob demands proof for $a(F_2)[x_1 \mapsto r_1] = a_2 \bmod p_i$
  3. Alice sends polynomial $\rho_2(x_2)$
  4. For $Q_2 = \exists$ Bob checks $a_2 \equiv \rho_2(0) + \rho_2(1) \bmod p_i$
  5. For $Q_2 = \forall$ Bob checks $a_2 \equiv \rho_2(0) \cdot \rho_2(1) \bmod p_i$
  6. Bob randomly selects $0 \leq r_2 < p_i$, shares it with Alice and computes $\rho_2(r_2) \bmod p_i$
  7. write $a(G_2)[x_1 \mapsto r_1, x_2 \mapsto r_2]$ as $b + c \cdot a(F_3)[x_1 \mapsto r_1, x_2 \mapsto r_2]$ with $F_3$ subformula of $G_2$ starting with first quantifier
  8. Bob computes $0 \leq b, c < p_i$
  9. Bob accepts if $c = 0$ and $\rho_2(r_2) = b$
  10. otherwise Bob computes $a_3 = (\rho_2(r_2) - b) \cdot c^{-1} \bmod p_i$

Proof (6/7)

- other rounds accordingly

Proof (6/7)

- other rounds accordingly
- Bob accepts $F \in \mathrm{QBF}$ with probability 1

Proof (6/7)

- other rounds accordingly
- Bob accepts $F \in \text{QBF}$ with probability 1
- let $F \notin \text{QBF}$ and thus $a(F) \equiv 0 \bmod p_i$
- what is probability that Bob accepts? Suppose Bob does

Proof (6/7)

- other rounds accordingly
- Bob accepts $F \in \text{QBF}$ with probability 1
- let $F \notin \text{QBF}$ and thus $a(F) \equiv 0 \bmod p_i$
- what is probability that Bob accepts? Suppose Bob does
- Alice sends $a_1 \not\equiv a(F_1) \equiv 0 \bmod p_i$ in round 1

Proof (6/7)

- other rounds accordingly
- Bob accepts $F \in \text{QBF}$ with probability 1
- let $F \notin \text{QBF}$ and thus $a(F) \equiv 0 \bmod p_i$
- what is probability that Bob accepts? Suppose Bob does
- Alice sends $a_1 \not\equiv a(F_1) \equiv 0 \bmod p_i$ in round 1
- Bob checks $a_1 \equiv \rho_1(0) \otimes \rho_1(1) \bmod p_i$,
  so the sent polynomial $\rho_1(x_1) \neq a(G_1)$ is wrong

Proof (6/7)

- other rounds accordingly
- Bob accepts $F \in \text{QBF}$ with probability 1
- let $F \notin \text{QBF}$ and thus $a(F) \equiv 0 \bmod p_i$
- what is probability that Bob accepts? Suppose Bob does
- Alice sends $a_1 \not\equiv a(F_1) \equiv 0 \bmod p_i$ in round 1
- Bob checks $a_1 \equiv \rho_1(0) \otimes \rho_1(1) \bmod p_i$,
  so the sent polynomial $\rho_1(x_1) \neq a(G_1)$ is wrong
- $\rho_1(x_1) - a(G_1)$ has degree at most $2n$ and thus at most $2n$ roots
- $\rho_1(r_1) = a(G_1)[x_1 \mapsto r_1]$ holds for at most $2n$ values $0 \leq r_1 < p_i$

$$\text{Prob}\Big[\rho_1(r_1) = a(G_1)[x_1 \mapsto r_1]\Big] \leq \frac{2n}{2^n}$$

for uniform $r_1$ since $p_i \geq 2^n$

<u>Proof</u> (6/7)

- other rounds accordingly
- Bob accepts $F \in \mathrm{QBF}$ with probability 1
- let $F \notin \mathrm{QBF}$ and thus $a(F) \equiv 0 \bmod p_i$
- what is probability that Bob accepts? Suppose Bob does
- Alice sends $a_1 \not\equiv a(F_1) \equiv 0 \bmod p_i$ in round 1
- Bob checks $a_1 \equiv \rho_1(0) \otimes \rho_1(1) \bmod p_i$,
  so the sent polynomial $\rho_1(x_1) \neq a(G_1)$ is wrong
- $\rho_1(x_1) - a(G_1)$ has degree at most $2n$ and thus at most $2n$ roots
- $\rho_1(r_1) = a(G_1)[x_1 \mapsto r_1]$ holds for at most $2n$ values $0 \leq r_1 < p_i$

$$\mathrm{Prob}\Big[\rho_1(r_1) = a(G_1)[x_1 \mapsto r_1]\Big] \leq \frac{2n}{2^n}$$

  for uniform $r_1$ since $p_i \geq 2^n$
- $\rho_1(r_1) = a(G_1)[x_1 \mapsto r_1] \iff a_2 = a(F_2)[x_1 \mapsto r_1]$ provided $c \neq 0$

Proof (7/7)

- $\text{Prob}\big[\rho_1(r_1) \neq a(G_1)[x_1 \mapsto r_1]\big] \geq 1 - \frac{2n}{2^n}$ at start of round 2
- argument repeats with demand for proof of $a_2 = a(F_2)[x_1 \mapsto r_1]$

Proof (7/7)

- $\text{Prob}\left[\rho_1(r_1) \neq a(G_1)[x_1 \mapsto r_1]\right] \geq 1 - \frac{2n}{2^n}$ at start of round 2
- argument repeats with demand for proof of $a_2 = a(F_2)[x_1 \mapsto r_1]$
- at most $\ell$ rounds
- probability of correct answer "$F \notin \text{QBF}$" is at least

$$\left(1 - \frac{2n}{2^n}\right)^\ell \geq \left(1 - \frac{2n}{2^n}\right)^n \geq 1 - n \cdot \frac{2n}{2^n}$$

since $(1-x)^n \geq 1 - nx$ for $0 \leq x \leq 1$

Proof (7/7)

- $\text{Prob}\big[\rho_1(r_1) \neq a(G_1)[x_1 \mapsto r_1]\big] \geq 1 - \frac{2n}{2^n}$ at start of round 2
- argument repeats with demand for proof of $a_2 = a(F_2)[x_1 \mapsto r_1]$
- at most $\ell$ rounds
- probability of correct answer "$F \notin \text{QBF}$" is at least

$$\left(1 - \frac{2n}{2^n}\right)^{\ell} \geq \left(1 - \frac{2n}{2^n}\right)^{n} \geq 1 - n \cdot \frac{2n}{2^n}$$

  since $(1 - x)^n \geq 1 - nx$ for $0 \leq x \leq 1$
- probability of wrong answer "$F \in \text{QBF}$" is at most $\frac{2n^2}{2^n}$

Proof (7/7)

- $\text{Prob}\left[\rho_1(r_1) \neq a(G_1)[x_1 \mapsto r_1]\right] \geq 1 - \frac{2n}{2^n}$ at start of round 2
- argument repeats with demand for proof of $a_2 = a(F_2)[x_1 \mapsto r_1]$
- at most $\ell$ rounds
- probability of correct answer "$F \notin \text{QBF}$" is at least

$$\left(1 - \frac{2n}{2^n}\right)^\ell \geq \left(1 - \frac{2n}{2^n}\right)^n \geq 1 - n \cdot \frac{2n}{2^n}$$

  since $(1-x)^n \geq 1 - nx$ for $0 \leq x \leq 1$
- probability of wrong answer "$F \in \text{QBF}$" is at most $\frac{2n^2}{2^n}$
- rerun of protocol lowers it to $\left(\frac{2n^2}{2^n}\right)^2 = \frac{4n^4}{2^{2n}} < 2^{-n}$ for large $n$ $\square$