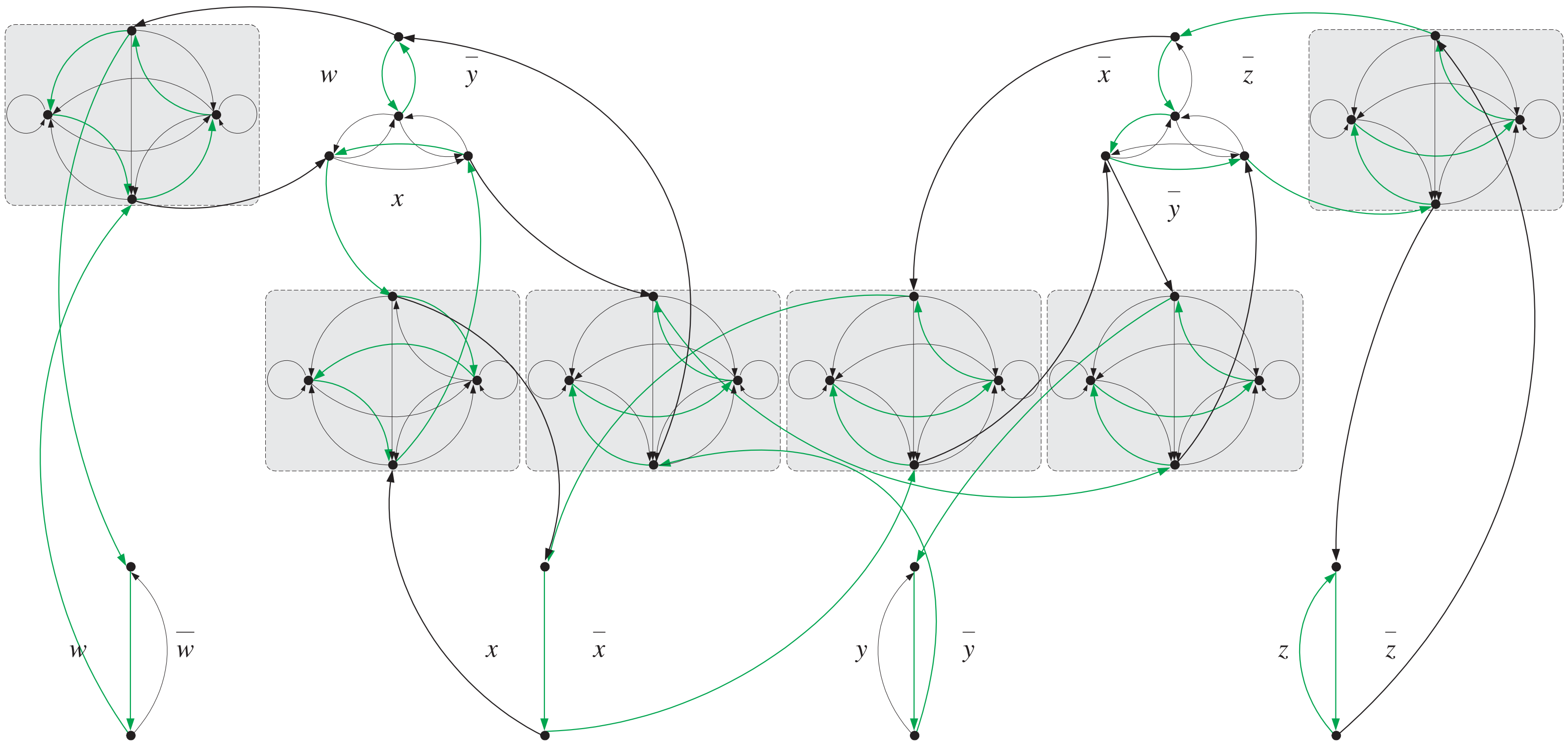
# "$w = 1, x = 0, y = 0, z = 1$" Adds $4^6$ to Cycle Count



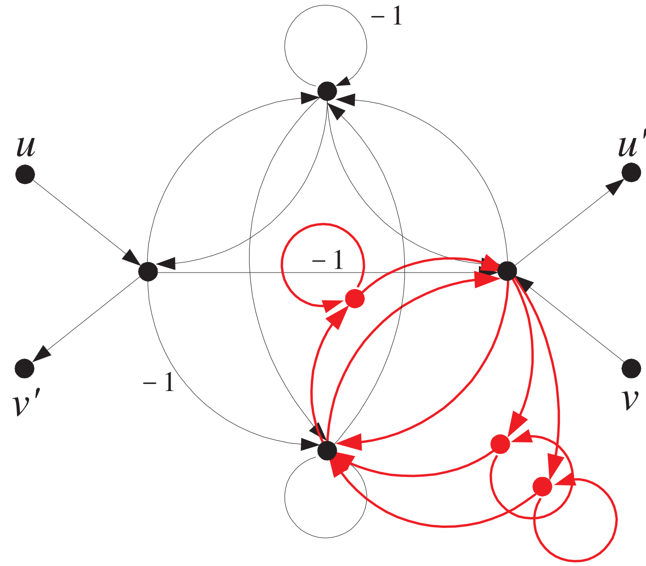
## The Proof (continued)

- We are almost done.
- The weighted directed graph $H$ needs to be *efficiently* replaced by some unweighted graph $G$.
- Furthermore, knowing $\#G$ should enable us to calculate $\#H$ *efficiently*.
  - This done, $\#\phi$ will have been Turing-reducible to $\#G$.[a]
- We proceed to construct this graph $G$.

[a] By way of $\#H$ of course.

## The Proof: Construction of $G$ (continued)

- Replace edges with weights 2 and 3 as follows (note that the graph cannot have parallel edges):




- The cycle count $\#H$ remains *unchanged*.

## The Proof: Construction of $G$ (continued)

- We move on to edges with weight $-1$.
- First, we count the number of nodes, $M$.
- Each clause gadget contains 4 nodes (p. 653), and there are $m$ of them (one per clause).
- Each revised XOR gadget contains 7 nodes (p. 672), and there are $3m$ of them (one per literal).
- Each choice gadget contains 2 nodes (p. 664), and there are $n \leq 3m$ of them (one per variable).
- So
$$M \leq 4m + 21m + 6m = 31m.$$

## The Proof: Construction of $G$ (continued)

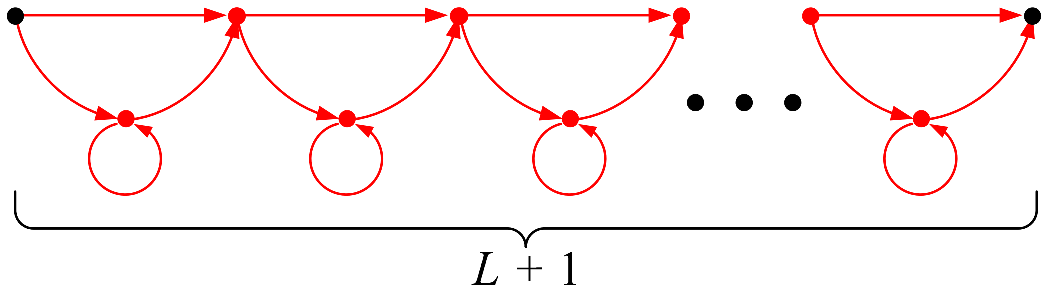
- $\#H \leq 2^L$ for some $L = O(m \log m)$.
  - The maximum absolute value of the edge weight is 1.
  - Hence each term in the permanent is at most 1.
  - There are $M! \leq (31m)!$ terms.
  - Hence

$$\begin{aligned} \#H &\leq \sqrt{2\pi(31m)} \left(\frac{31m}{e}\right)^{31m} e^{\frac{1}{12\times(31m)}} \\ &= 2^{O(m \log m)} \end{aligned} \tag{10}$$

by a refined Stirling's formula.

## The Proof: Construction of $G$ (continued)

- Replace each edge with weight $-1$ with the following:




- Each increases the number of cycle covers $2^{L+1}$-fold.
- The desired unweighted $G$ has been obtained.

## The Proof (continued)

- $\#G$ equals $\#H$ after replacing each appearance $-1$ in $\#H$ with $2^{L+1}$:

$$\begin{aligned}
\#H &= \cdots + \overbrace{(-1) \cdot 1 \cdots\cdots 1}^{\text{a cycle cover}} + \cdots, \\
\#G &= \cdots + \overbrace{2^{L+1} \cdot 1 \cdots\cdots 1}^{\text{a cycle cover}} + \cdots.
\end{aligned}$$

- Let $\#G = \sum_{i=0}^{n} a_i \times (2^{L+1})^i$, where $0 \leq a_i < 2^{L+1}$.
- As $\#H \leq 2^L$ even if we replace $-1$ by 1 (p. 674), each $a_i$ equals the number of cycle covers with $i$ edges of weight $-1$.

## The Proof (concluded)

- We conclude that

$$\#H = a_0 - a_1 + a_2 - \cdots + (-1)^n a_n,$$

indeed easily computable from $\#G$.

- We know $\#H = 4^{3m} \times \#\phi$ (p. 669).
- So

$$\#\phi = \frac{a_0 - a_1 + a_2 - \cdots + (-1)^n a_n}{4^{3m}}.$$

  - More succinctly,

$$\#\phi = \frac{\#G \bmod (2^{L+1} + 1)}{4^{3m}}.$$

*Polynomial Space*

# PSPACE and Games

- Given a boolean expression $\phi$ in CNF with boolean variables $x_1, x_2, \ldots, x_n$, is it true that $\exists x_1 \forall x_2 \cdots Q_n x_n \phi$?
- This is called **quantified satisfiability** or QSAT.
- This problem is like a two-person game: $\exists$ and $\forall$ are the two players.
- We ask then is there a winning strategy for $\exists$?
- QSAT Is PSPACE-Complete[a]

[a] Stockmeyer and Meyer (1973).

## IP and PSPACE

- We next prove that coNP $\subseteq$ IP.
- Shamir in 1990 proved that IP equals PSPACE using similar ideas (p. 709).

# Interactive Proof for Boolean Unsatisfiability

- Like GRAPH NONISOMORPHISM (p. 538), it is not clear how to construct a short certificate for UNSAT.
- But with interaction and randomization, we shall present an interactive proof for UNSAT.
- A 3SAT formula is a conjunction of disjunctions of at most three literals.
- For any unsatisfiable 3SAT formula $\phi(x_1, x_2, \ldots, x_n)$, there is an interactive proof for the fact that it is unsatisfiable.
- Therefore, coNP $\subseteq$ IP.

# Arithmetization of Boolean Formulas

The idea is to arithmetize the boolean formula.

- T $\rightarrow$ positive integer
- F $\rightarrow 0$
- $x_i \rightarrow x_i$
- $\neg x_i \rightarrow 1 - x_i$
- $\vee \rightarrow +$
- $\wedge \rightarrow \times$
- $\phi(x_1, x_2, \ldots, x_n) \rightarrow \Phi(x_1, x_2, \ldots, x_n)$

## The Arithmetized Version

- A boolean formula is transformed into a multivariate polynomial $\Phi$.
- It is easy to verify that $\phi$ is unsatisfiable if and only if

$$\sum_{x_1=0,1} \sum_{x_2=0,1} \cdots \sum_{x_n=0,1} \Phi(x_1, x_2, \ldots, x_n) = 0.$$

- But the above seems to require exponential time.
- We turn to more intricate methods.

# Choosing the Field

- Suppose $\phi$ has $m$ clauses of length three each.
- Then $\Phi(x_1, x_2, \ldots, x_n) \leq 3^m$ for any truth assignment $(x_1, x_2, \ldots, x_n)$.
- Because there are at most $2^n$ truth assignments,

$$\sum_{x_1=0,1} \sum_{x_2=0,1} \cdots \sum_{x_n=0,1} \Phi(x_1, x_2, \ldots, x_n) \leq 2^n 3^m.$$

# Choosing the Field (concluded)

- By choosing a prime $q > 2^n 3^m$ and working modulo this prime, proving unsatisfiability reduces to proving that

$$\sum_{x_1=0,1} \sum_{x_2=0,1} \cdots \sum_{x_n=0,1} \Phi(x_1, x_2, \ldots, x_n) \equiv 0 \bmod q. \quad (11)$$

- Working under a *finite* field allows us to uniformly select a random element in the field.

## Binding Peggy

- Peggy has to find a sequence of polynomials that satisfy a number of restrictions.
- The restrictions are imposed by Victor: After receiving a polynomial from Peggy, Victor sets a new restriction for the next polynomial in the sequence.
- These restrictions guarantee that if $\phi$ is unsatisfiable, such a sequence can always be found.
- However, if $\phi$ is not unsatisfiable, any Peggy has only a small probability of finding such a sequence.
  - The probability is taken over Victor's coin tosses.

## The Algorithm

Peggy and Victor both arithmetize $\phi$ to obtain $\Phi$;
Peggy picks a prime $q > 2^n 3^m$ and sends it to Victor;
Victor rejects and stops if $q$ is not a prime;
Victor sets $v_0 = 0$;
**for** $i = 1, 2, \ldots, n$ **do**
    Peggy calculates $P_i^*(z) = \sum_{x_{i+1}=0,1} \cdots \sum_{x_n=0,1} \Phi(r_1, \ldots, r_{i-1}, z, x_{i+1}, \ldots, x_n)$;
    Peggy sends $P_i^*(z)$ to Victor;
    Victor rejects and stops if $P_i^*(0) + P_i^*(1) \not\equiv v_{i-1} \bmod q$ or $P_i^*(z)$'s degree exceeds $m$; $\{P_i^*(z)$ has at most $m$ clauses.$\}$
    Victor uniformly picks $r_i \in Z_q$ and calculates $v_i = P_i^*(r_i)$;
    Victor sends $r_i$ to Peggy;
**end for**
Victor accepts iff $\Phi(r_1, r_2, \ldots, r_n) \equiv v_n \bmod q$;

## Comments

- The following invariant is maintained by the algorithm:

$$P_i^*(0) + P_i^*(1) \equiv P_{i-1}^*(r_{i-1}) \bmod q \tag{12}$$

  for $1 \leq i \leq n$.

  - $P_i^*(0) + P_i^*(1)$ equals $\sum_{x_i=0,1} \cdots \sum_{x_n=0,1} \Phi(r_1, \ldots, r_{i-1}, x_i, x_{i+1}, \ldots, x_n)$ modulo $q$.
  - The above equals $P_{i-1}^*(r_{i-1}) \bmod q$ by definition.

## Comments (concluded)

- The computation of $v_1, v_2, \ldots, v_n$ must rely on Peggy's supplied polynomials as Victor does not have the power to carry out the exponential-time calculations.
- But $\Phi(r_1, r_2, \ldots, r_n)$ in Step 12 is computed without relying on Peggy.

## Completeness

- Suppose $\phi$ is unsatisfiable.
- For $i \geq 1$, by Eq. (12) on p. 688,

$$\begin{aligned}
& P_i^*(0) + P_i^*(1) \\
= & \sum_{x_i=0,1} \sum_{x_{i+1}=0,1} \cdots \sum_{x_n=0,1} \Phi(r_1, \ldots, r_{i-1}, x_i, x_{i+1}, \ldots, x_n) \\
= & P_{i-1}^*(r_{i-1}) \\
\equiv & v_{i-1} \bmod q.
\end{aligned}$$

# Completeness (concluded)

- In particular at $i = 1$, because $\phi$ is unsatisfiable, we have

$$\begin{aligned} P_1^*(0) + P_1^*(1) &= \sum_{x_1=0,1} \cdots \sum_{x_n=0,1} \Phi(x_1, \ldots, x_n) \\ &\equiv v_0 \\ &= 0 \bmod q. \end{aligned}$$

- Finally, $v_n = P_n^*(r_n) = \Phi(r_1, r_2, \ldots, r_n)$.
- Because all the tests by Victor will pass, Victor will accept $\phi$.

## Soundness

- Suppose $\phi$ is not unsatisfiable.
- Victor will reject after an honest Peggy sends $P_1^*(z)$.
  - $P_1^*(z) = \sum_{x_2=0,1} \cdots \sum_{x_n=0,1} \Phi(z, x_2, \ldots, x_n)$.
  - So

$$\begin{aligned} & P_1^*(0) + P_1^*(1) \\ = & \sum_{x_1=0,1} \sum_{x_2=0,1} \cdots \sum_{x_n=0,1} \Phi(x_1, x_2, \ldots, x_n) \\ \not\equiv & \ 0 \bmod q \end{aligned}$$

    by Eq. (11) on p. 685.
  - But $v_0 = 0$.

## Soundness (continued)

- We will show that if Peggy is dishonest in one round (by sending a polynomial other than $P_i^*(z)$), then with high probability she must be dishonest in the next round, too.
- In the last round (Step 12), her dishonesty is exposed.

## Soundness (continued)

- Let $P_i(z)$ represent the polynomial sent by Peggy in place of $P_i^*(z)$.
- Victor calculates $v_i = P_i(r_i) \bmod p$.
- In order to deceive Victor in the next round, round $i+1$, Peggy must use $r_1, r_2, \ldots, r_i$ to find a $P_{i+1}(z)$ of degree at most $m$ such that

$$P_{i+1}(0) + P_{i+1}(1) \equiv v_i \bmod q$$

(see Step 8 of the algorithm on p. 687).

- And so on to the end, except that Peggy has no control over Step 12.

## A Key Claim

**Lemma 88** *If* $P_i^*(0) + P_i^*(1) \not\equiv v_{i-1} \bmod q$*, then either Victor rejects in the* $i$*th round, or* $P_i^*(r_i) \not\equiv v_i \bmod q$ *with probability at least* $1 - (m/q)$*, where the probability is taken over Victor's choices of* $r_i$*.*

- Think of $P_i^*(r_i)$ as the $v_i$ that Victor *should be* computing if Peggy were honest.
- But Victor actually calculates $P_i(z)$ as $v_i$ (Peggy claims $P_i(z)$ is $P_i^*(z)$):

$$v_i = P_i(r_i) \bmod q.$$

- What Victor can do is to check for consistencies.

## The Proof of Lemma 88 (continued)

- If Peggy sends a $P_i(z)$ which equals $P_i^*(z)$, then

$$P_i(0) + P_i(1) = P_i^*(0) + P_i^*(1) \not\equiv v_{i-1} \bmod q,$$

  and Victor rejects immediately.

- Suppose Peggy sends a $P_i(z)$ different from $P_i^*(z)$.

- If $P_i(z)$ does not pass Victor's test

$$P_i(0) + P_i(1) \equiv v_{i-1} \bmod q? \tag{13}$$

  then Victor rejects and we are done, too.

## The Proof of Lemma 88 (concluded)

- Finally, assume $P_i(z)$ passes the test (13) on p. 696.
- $P_i(z) - P_i^*(z) \not\equiv 0$ is a polynomial of degree at most $m$.
- Hence equation $P_i(z) - P_i^*(z) \equiv 0 \bmod q$ has at most $m$ roots $r \in Z_q$, i.e.,

$$P_i^*(r) \equiv P_i(r) \bmod q.$$

- Hence Victor will pick one of these as his $r_i$ so that

$$P_i^*(r_i) \equiv P_i(r_i) \equiv v_i \bmod q$$

with probability at most $m/q$.

## Soundness (continued)

- Suppose Victor does not reject in any of the first $n$ rounds.
- As $\phi$ is not unsatisfiable,

$$P_1^*(0) + P_1^*(1) \not\equiv v_0 \bmod q.$$

- By Lemma 88 (p. 695) and the fact that Victor does not reject, we have $P_1^*(r_1) \not\equiv v_1 \bmod q$ with probability at least $1 - (m/q)$.
- Now by Eq. (12) on p. 688,

$$P_1^*(r_1) = P_2^*(0) + P_2^*(1) \not\equiv v_1 \bmod q.$$

## Soundness (concluded)

- Iterating on this procedure, we eventually arrive at

$$P_n^*(r_n) \not\equiv v_n \bmod q$$

with probability at least $(1 - m/q)^n$.

- As $P_n^*(r_n) = \Phi(r_1, r_2, \ldots, r_n)$, Victor's last test at Step 12 fails and he rejects.
- Altogether, Victor rejects with probability at least

$$[\,1 - (m/q)\,]^n > 1 - (nm/q) > 2/3 \tag{14}$$

because $q > 2^n 3^m$.

# An Example

- $(x_1 \lor x_2 \lor x_3) \land (x_1 \lor \neg x_2 \lor \neg x_3)$.
- The above is satisfied by assigning true to $x_1$.
- The arithmetized formula is

  $$\Phi(x_1, x_2, x_3) = (x_1 + x_2 + x_3) \times [\, x_1 + (1 - x_2) + (1 - x_3) \,].$$

- Indeed, $\sum_{x_1=0,1} \sum_{x_2=0,1} \sum_{x_3=0,1} \Phi(x_1, x_2, x_3) = 16 \neq 0$.
- We have $n = 3$ and $m = 2$.
- A prime $q$ that satisfies $q > 2^3 \times 3^2 = 72$ is 73.

## An Example (continued)

- The table below is an execution of the algorithm in $Z_{73}$ *when Peggy follows the protocol.*

| $i$ | $P_i^*(z)$ | $P_i^*(0) + P_i^*(1)$ | $= v_{i-1}$? | $r_i$ | $v_i$ |
|---|---|---|---|---|---|
| 0 | | | | | 0 |
| 1 | $4z^2 + 8z + 2$ | 16 | no | | |

- Victor therefore rejects $\phi$ early on at $i = 1$.

## An Example (continued)

- Now suppose Peggy does not follow the protocol.
- In order to deceive Victor, she comes up with fake polynomials $P_i(z)$ from $i = 1$.
- The table below is an execution of the algorithm.

| $i$ | $P_i(z)$ | $P_i(0) + P_i(1)$ | $= v_{i-1}$? | $r_i$ | $v_i$ |
|---|---|---|---|---|---|
| 0 | | | | | 0 |
| 1 | $8z^2 + 11z + 27$ | 0 | yes | 2 | 35 |
| 2 | $z^2 + 8z + 13$ | 35 | yes | 3 | 46 |
| 3 | $3z^2 + z + 21$ | 46 | yes | $r_3$ | $P_3(r_3)$ |

## An Example (concluded)

- Victor has been satisfied up to round 3.
- Finally at Step 12, Victor checks if

$$\Phi(2, 3, r_3) \equiv P_3(r_3) \bmod 73.$$

- It can be verified that the only choices of $r_3 \in \{0, 1, \ldots, 72\}$ that can mislead Victor are 31 and 59.
- The probability of that happening is only 2/73.[a]

[a] Ms. Ching-Ju Lin (R92922038) on January 7, 2004, pointed out an error in an earlier calculation.

## An Example

- $(x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2)$.
- The above is unsatisfiable.
- The arithmetized formula is

  $$\Phi(x_1, x_2) = (x_1 + x_2) \times (x_1 + 1 - x_2) \times (1 - x_1 + x_2) \times (2 - x_1 - x_2).$$

- Because $\Phi(x_1, x_2) = 0$ for any *boolean* assignment $\{0, 1\}^2$ to $(x_1, x_2)$, certainly

  $$\sum_{x_1=0,1} \sum_{x_2=0,1} \Phi(x_1, x_2) = 0.$$

- With $n = 2$ and $m = 4$, a prime $q$ that satisfies $q > 2^2 \times 3^4 = 4 \times 81 = 324$ is 331.

# An Example (concluded)

- The table below is an execution of the algorithm in $Z_{331}$.

| $i$ | $P_i^*(z)$ | $P_i^*(0) + P_i^*(1)$ | $= v_{i-1}$? | $r_i$ | $v_i$ |
|---|---|---|---|---|---|
| 0 | | | | | 0 |
| 1 | $z(z+1)(1-z)(2-z)$ $+(z+1)z(2-z)(1-z)$ | 0 | yes | 10 | 283 |
| 2 | $(10+z)\times(11-z)$ $\times(-9+z)\times(-8-z)$ | 283 | yes | 5 | 46 |

- Victor calculates $\Phi(10, 5) \equiv 46 \bmod 331$.
- As it equals $v_2 = 46$, Victor accepts $\phi$ as unsatisfiable.

# Objections to the Soundness Proof?[a]

- Based on the steps required of a cheating Peggy on p. 694, why must we go through so many rounds (in fact, $n$ rounds)?
- Why not just go directly to round $n$:
  - Victor sends $r_1, r_2, \ldots, r_{n-1}$ to Peggy.
  - Peggy returns with a (claimed) $P_n^*(z)$.
  - Victor accepts if and only if $\Phi(r_1, r_2, \ldots, r_{n-1}, r_n) \equiv P_n^*(r_n) \bmod q$ for a random $r_n \in Z_q$.

[a]Contributed by Ms. Emily Hou (D89011) and Mr. Pai-Hsuen Chen (R90008) on January 2, 2002.

## Objections to the Soundness Proof? (continued)

- Let us analyze the compressed proposal when $\phi$ is satisfiable.
- To succeed in foiling Victor, Peggy must find a polynomial $P_n(z)$ of degree $m$ such that

$$\Phi(r_1, r_2, \ldots, r_{n-1}, z) \equiv P_n(z) \bmod q.$$

- But this she is able to do: Just give the verifier the polynomial $\Phi(r_1, r_2, \ldots, r_{n-1}, z)$!
- What has happened?

# Objections to the Soundness Proof? (concluded)

- You need the intermediate rounds to "tie" Peggy up with a chain of claims.
- In the original algorithm on p. 687, for example, $P_n(z)$ is bound by the equality $P_n(0) + P_n(1) \equiv v_{n-1} \bmod q$ in Step 8.
- That $v_{n-1}$ is in turn derived by an earlier polynomial $P_{n-1}(z)$, which is in turn bound by $P_{n-1}(0) + P_{n-1}(1) \equiv v_{n-2} \bmod q$, and so on.

## Shamir's Theorem[a]

**Theorem 89** *$IP = PSPACE$.*

- We first sketch the proof for IP $\subseteq$ PSPACE.
- Without loss of generality, assume:
  - If $x \in L$, then the probability that $x$ is accepted by the verifier is at least $3/4$.
  - If $x \notin L$, then the probability that $x$ is accepted by the verifier with *any* prover is at most $1/4$.

[a] Shamir (1990).

## The Proof (continued)

- Now we track down every possible message exchange based on random choices by the verifier and all possible messages generated by the prover.
- Use recursion to calculate

$$\text{prob}[\,\text{verifier accepts } x\,]$$

as

$$\max_P \text{prob}[\,(V, P) \text{ accepts } x\,].$$

- If this value is at least 3/4, then acce[t $x$; otherwise, reject $x$.

# The Proof (continued)

- To prove PSPACE $\subseteq$ IP, we next prove that QSAT is in IP.
- We do so by describing an interactive protocol that decides QSAT.
- Suppose Alice and Bob are given

$$\begin{aligned}\phi &= \forall x \exists y (x \lor y) \land \forall z [\, (x \land z) \lor (y \land \neg z)\,] \\ &\quad \lor \exists w [\, z \lor (y \land \neg w)\,].\end{aligned}$$

- As above, we assume no occurrence of a variable is separated by more than one $\forall$ from its point of quantification.

# Proof of Theorem (continued)

- We also assume that $\neg$ is applied only to variables, not subexpressions.
- We now arithmetize $\phi$ as before except:
  - 1 means true.
  - $\neg x \to 1 - x$.
    * It is the standard representation on p. 134.
  - $\exists x \to \sum_{x=0,1}$.
  - $\forall x \to \prod_{x=0,1}$.
- Alice tries to convince Bob that this arithmetization of $\phi$ is nonzero.

## Proof of Theorem (continued)

- Our $\phi$ becomes

$$
\begin{aligned}
A_\phi &= \prod_{x=0}^{1}\sum_{y=0}^{1}\{(x+y)\cdot\prod_{z=0}^{1}[(x\cdot z+y\cdot(1-z))\\
&\quad +\sum_{w=0}^{1}(z+y\cdot(1-w))]\}.
\end{aligned}
$$

- Call it a $\sum - \prod$ expression.
- $A_\phi$ is a number; it equals 96 here.

## Proof of Theorem (continued)

- As before, $\phi$ is true if and only if $A_\phi > 0$.
- In fact, more is true.
- For any $\phi$ and any truth assignment to its free variables:
  - If $\phi$ is true, then $A_\phi > 0$ under the corresponding 0-1 assignment.
  - If $\phi$ is false, then $A_\phi = 0$.
- So Alice only has to convince Bob that $A_\phi > 0$.

## Proof of Theorem (continued)

- The trouble is that $A_\phi$ evaluated can be exponential in length.
- Fortunately, it can be shown that if expression $A_\phi$ of length $n$ is nonzero, then there is a prime $p$ between $2^n$ and $2^{3n}$ such that $A_\phi \neq 0 \bmod p$.
- So Alice only has to convince Bob that $A_\phi \neq 0$ under mod$p$.
- The protocol starts with Alice sending Bob $p$ (assume $p = 13$) and its primality certificate.

# Proof of Theorem (continued)

- Now Alice sends Bob $A_\phi$ mod $p$, which is

$$\boldsymbol{a} = 96 \bmod 13 = 5.$$

- Each stage starts with the following:
  - A $\sum - \prod$ expression $A$, with a leading $\sum_x$ or $\prod_x$.
  - $A$'s alleged value $a$ mod $p$, supplied by Alice.
- If the first $\sum$ or $\prod$ is deleted, the result is a polynomial in $x$, called $A'(x)$.
- Bob demands from Alice the coefficients of $A'(x)$.
- Trouble occurs if the degree of $A'(x)$ is exponential in $n$.

## Proof of Theorem (continued)

- Luckily, $\deg(A'(x)) \leq 2n$.
  - No occurrence of a variable is separated by more than one $\forall$ from its point of quantification.
  - So $A'(x)$ has only one $\prod$ symbol.
  - Other $\prod$s are over quantities not related to $x$, hence purely numerical.
  - Symbols other than $\prod$ can only increase the degree of $A'(x)$ by at most one $(x \cdot x \cdots)$.
  - For example, $\sum_y (x + y) \prod_z (x + \sum_w (x \cdot w))$.
- So Alice has no problem transmitting $A'(x)$ to Bob.

## Proof of Theorem (continued)

- $A'(x) = 2x^2 + 8x + 6$.
- Bob verifies that $A'(0) \cdot A'(1) = 5 \bmod 13$.
- Indeed $A'(0) \cdot A'(1) = 6 \cdot 16 = 5 \bmod 13$.
- So far $A'(x)$ is consistent with the alleged value 5.
- Bob deletes the leading $\prod_x$.
- The *free variable* $x$ in the resulting expression prevents it from being an evaluation problem.

# Proof of Theorem (continued)

- So Bob replaces $x$ with a random number mod13, say 9:

$$\sum_{y=0}^{1} \left\{ (9+y) \cdot \prod_{z=0}^{1} \left[ (9 \cdot z + y \cdot (1-z)) + \sum_{w=0}^{1} (z + y \cdot (1-w)) \right] \right\}.$$

- The above equals

$$\boldsymbol{a} = A'(9) = 2 \cdot 9^2 + 8 \cdot 9 + 6 = 6 \bmod 13.$$

- Bob sends 9 to Alice.

## Proof of Theorem (continued)

- In the new stage, Alice evaluates

$$A'(y) = 2y^3 + y^2 + 3y$$

  after substituting $x = 9$ and sends it to Bob.

- Bob checks that $A'(0) + A'(1) = 6 \bmod 13$.
- Indeed $0 + 6 = 6 \bmod 13$.
- Bob deletes the leading $\sum_y$.
- Bob replaces $y$ with a random number mod13, say 3:

$$(9+3) \cdot \prod_{z=0}^{1} \left\{ [\, 9 \cdot z + 3 \cdot (1 - z) \,] + \sum_{w=0}^{1} [\, z + 3 \cdot (1 - w) \,] \right\}.$$

## Proof of Theorem (continued)

- The above should equal $A'(3) = 2 \cdot 3^2 + 3^2 + 3 \cdot 3 = 7 \bmod 13$.
- So

$$A = \prod_{z=0}^{1} \{ [\, 9 \cdot z + 3 \cdot (1 - z) \,] + \sum_{w=0}^{1} [\, z + 3 \cdot (1 - w) \,] \}$$

  should equal

$$\boldsymbol{a} = 12^{-1} \cdot 7 = 12 \cdot 7 = 6 \bmod 13.$$

- Bob sends 3 to Alice.

## Proof of Theorem (continued)

- In the new stage, Alice evaluates

$$A'(z) = 8z + 6$$

after substituting $y = 3$ and sends it to Bob.

- Bob checks that $A'(0) \cdot A'(1) = 6 \bmod 13$.
- Indeed $6 \cdot 14 = 6 \bmod 13$.
- Bob deletes the leading $\prod_z$.
- Bob replaces $z$ with a random number mod13, say 7:

$$[\,9 \cdot 7 + 3 \cdot (1 - 7)\,] + \sum_{w=0}^{1} [\,7 + 3 \cdot (1 - w)\,].$$

## Proof of Theorem (continued)

- The above should equal $A'(7) = 8 \cdot 7 + 6 = 10 \bmod 13$.
- So

$$A = \sum_{w=0}^{1} [\, z + 3 \cdot (1 - w) \,] \tag{15}$$

  should equal

$$\boldsymbol{a} = 10 - [\, 9 \cdot 7 + 3 \cdot (1 - 7) \,] = 10 - 45 = 4 \bmod 13.$$

- Bob sends 7 to Alice.

## Proof of Theorem (continued)

- In the new stage, Alice evaluates

$$A'(w) = 10 - 3w$$

  after substituting $z = 7$ and sends it to Bob.

- Bob checks that $A'(0) + A'(1) = 4 \bmod 13$.
- Indeed $10 + 7 = 4 \bmod 13$.
- Now there are no more $\sum$s and $\prod$s.
- Bob checks if $A'(w)$ is indeed as claimed by using (15) with $z = 7$.
- It is, and Bob accepts $A_\phi \neq 0 \bmod 13$.

## Proof of Theorem (continued)

- Clearly, if $A_\phi > 0$, the protocol convinces Bob of this.
- We next show that if $A_\phi = 0$, then Bob will be cheated with only negligible probability.

**Lemma 90** *Suppose $A_\phi = 0$ and Alice claims a nonzero value $\boldsymbol{a}$. Then with probability $\geq (1 - \frac{2n}{2^n})^{i-1}$, the value of $\boldsymbol{a}$ claimed at the $i$th stage is wrong.*