The Proof: OR

- CC($\mathcal{X} \cup \mathcal{Y}$) is *equivalent to* the OR of CC($\mathcal{X}$) and CC($\mathcal{Y}$).
- Violations occur when $|\mathcal{X} \cup \mathcal{Y}| > M$.
- Such violations can be eliminated by using

$$\text{CC}(\text{pluck}(\mathcal{X} \cup \mathcal{Y}))$$

  as the approximate OR of CC($\mathcal{X}$) and CC($\mathcal{Y}$).
- We now count the numbers of errors this approximate OR makes on the positive and negative examples.

The Proof: OR (continued)

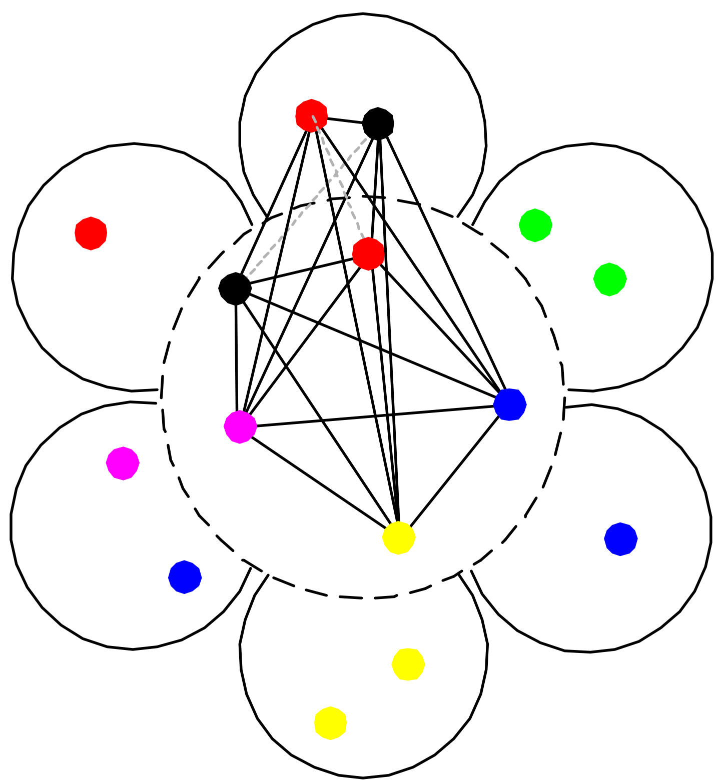
- CC(pluck($\mathcal{X} \cup \mathcal{Y}$)) *introduces* a **false negative** if a positive example makes either CC($\mathcal{X}$) or CC($\mathcal{Y}$) return true but makes CC(pluck($\mathcal{X} \cup \mathcal{Y}$)) return false.
- CC(pluck($\mathcal{X} \cup \mathcal{Y}$)) *introduces* a **false positive** if a negative example makes both CC($\mathcal{X}$) and CC($\mathcal{Y}$) return false but makes CC(pluck($\mathcal{X} \cup \mathcal{Y}$)) return true.
- How many false positives and false negatives are introduced by CC(pluck($\mathcal{X} \cup \mathcal{Y}$))?

The Number of False Positives

Lemma 76 CC(pluck($\mathcal{X} \cup \mathcal{Y}$)) *introduces at most* $\frac{2M}{p-1} 2^{-p}(k-1)^n$ *false positives.*

- Assume a plucking replaces the sunflower $\{Z_1, Z_2, \ldots, Z_p\}$ with its core Z.
- A false positive is *necessarily* a coloring such that:
  - There is a pair of identically colored nodes in each petal (and so both crude circuits return false).
  - But the core is all different colors.
    - This implies at least one node from each pair was plucked away.
- We now count the number of such colorings.

Proof of Lemma 76 (continued)



Proof of Lemma 76 (continued)

- Color nodes V at random with $k-1$ colors and let $R(X)$ denote the event that there are repeated colors in set X.
- Now $\text{prob}[R(Z_1) \wedge \cdots \wedge R(Z_p) \wedge \neg R(Z)]$ is at most

$$\begin{aligned} & \text{prob}[R(Z_1) \wedge \cdots \wedge R(Z_p) | \neg R(Z)] \\ = & \prod_{i=1}^{p} \text{prob}[R(Z_i) | \neg R(Z)] \leq \prod_{i=1}^{p} \text{prob}[R(Z_i)]. \quad (6) \end{aligned}$$

  - First equality holds because $R(Z_i)$ are independent given $\neg R(Z)$ as Z contains their only common nodes.
  - Last inequality holds as the likelihood of repetitions in Z_i decreases given no repetitions in $Z \subseteq Z_i$.

Proof of Lemma 76 (continued)

- Consider two nodes in Z_i.
- The probability that they have identical color is $\frac{1}{k-1}$.
- Now prob$[\, R(Z_i) \,] \leq \frac{\binom{|Z_i|}{2}}{k-1} \leq \frac{\binom{\ell}{2}}{k-1} \leq \frac{1}{2}$.
- So the probability that a random coloring is a new false positive is at most 2^{-p} by (6).
- As there are $(k-1)^n$ different colorings, each plucking introduces at most $2^{-p}(k-1)^n$ false positives.

Proof of Lemma 76 (concluded)

- $|\mathcal{X} \cup \mathcal{Y}| \leq 2M$.
- Each plucking reduces the number of sets by $p - 1$.
- Hence at most $\frac{2M}{p-1}$ pluckings occur in pluck$(\mathcal{X} \cup \mathcal{Y})$.
- At most $\frac{2M}{p-1} 2^{-p}(k-1)^n$ false positives are introduced.

The Number of False Negatives

Lemma 77 CC(pluck($\mathcal{X} \cup \mathcal{Y}$)) *introduces no false negatives.*

- Each plucking replaces a set in a crude circuit by a subset.
- This makes the test less stringent.
  - For each $Y \in \mathcal{X} \cup \mathcal{Y}$, there must exist at least one $X \in \text{pluck}(\mathcal{X} \cup \mathcal{Y})$ such that $X \subseteq Y$.
  - So if $Y \in \mathcal{X} \cup \mathcal{Y}$ is a clique, then pluck($\mathcal{X} \cup \mathcal{Y}$) also contains a clique in X.
- So plucking can only increase the number of accepted graphs.

The Proof: AND

- The approximate AND of crude circuits $\mathrm{CC}(\mathcal{X})$ and $\mathrm{CC}(\mathcal{Y})$ is

$$\mathrm{CC}(\mathrm{pluck}(\{X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y}, |X_i \cup Y_j| \leq \ell\})).$$

- We now count the numbers of errors this approximate AND makes on the positive and negative examples.

The Proof: AND (continued)

- The approximate AND *introduces* a **false negative** if a positive example makes both $\text{CC}(\mathcal{X})$ and $\text{CC}(\mathcal{Y})$ return true but makes the approximate AND return false.
- The approximate AND *introduces* a **false positive** if a negative example makes either $\text{CC}(\mathcal{X})$ or $\text{CC}(\mathcal{Y})$ return false but makes the approximate AND return true.
- How many false positives and false negatives are introduced by the approximate AND?

The Number of False Positives

Lemma 78 *The approximate* AND *introduces at most* $M^2 2^{-p}(k-1)^n$ *false positives.*

- $\text{CC}(\{X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y}\})$ introduces no false positives.
  - If $X_i \cup Y_j$ is a clique, both X_i and Y_j must be cliques, making both $\text{CC}(\mathcal{X})$ and $\text{CC}(\mathcal{Y})$ return true.
- $\text{CC}(\{X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y}, |X_i \cup Y_j| \leq \ell\})$ introduces no false positives because it is less stringent than above.

Proof of Lemma 78 (concluded)

- $|\{X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y}, |X_i \cup Y_j| \leq \ell\}| \leq M^2$.
- Each plucking reduces the number of sets by $p-1$.
- So pluck($\{X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y}, |X_i \cup Y_j| \leq \ell\}$) involves $< M^2/(p-1)$ pluckings.
- Each plucking introduces at most $2^{-p}(k-1)^n$ false positives by the proof of Lemma 76 (p. 482).
- The desired bound is

$$[\,M^2/(p-1)\,]\,2^{-p}(k-1)^n \leq M^2 2^{-p}(k-1)^n.$$

The Number of False Negatives

Lemma 79 *The approximate* AND *introduces at most* $M^2\binom{n-\ell-1}{k-\ell-1}$ *false negatives.*

- We follow the same three-step proof as before.
- $\mathrm{CC}(\{X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y}\})$ introduces no false negatives.
  - Suppose both $\mathrm{CC}(\mathcal{X})$ and $\mathrm{CC}(\mathcal{Y})$ accept a positive example with a clique of size k.
  - The clique must contain an $X_i \in \mathcal{X}$ and a $Y_j \in \mathcal{Y}$.
  - As it contains $X_i \cup Y_j$, the new circuit returns true.

Proof of Lemma 79 (concluded)

- $\mathrm{CC}(\{X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y}, |X_i \cup Y_j| \leq \ell\})$ introduces $\leq M^2 \binom{n-\ell-1}{k-\ell-1}$ false negatives.
  - Deletion of set Z larger than ℓ introduces false negatives which are cliques containing Z.
  - There are $\binom{n-|Z|}{k-|Z|}$ such cliques.
  - $\binom{n-|Z|}{k-|Z|} \leq \binom{n-\ell-1}{k-\ell-1}$ as $|Z| \geq \ell$.
  - There are at most M^2 such Zs.
- Plucking introduces no false negatives.

Two Summarizing Lemmas

From Lemmas 76 (p. 482) and 78 (p. 490), we have:

Lemma 80 *Each approximation step introduces at most* $M^2 2^{-p}(k-1)^n$ *false positives.*

From Lemmas 77 (p. 487) and 79 (p. 492), we have:

Lemma 81 *Each approximation step introduces at most* $M^2\binom{n-\ell-1}{k-\ell-1}$ *false negatives.*

The Proof (continued)

- The above two lemmas show that each approximation step introduce “few” false positives and false negatives.
- We next show that the resulting crude circuit has “a lot” of false positives or false negatives.

The Final Crude Circuit

Lemma 82 *Every final crude circuit either is identically false—thus wrong on all positive examples—or outputs true on at least half of the negative examples.*

- Suppose it is not identically false.
- By construction, it accepts at least those graphs that have a clique on some set X of nodes, with $|\, X \,| \leq \ell$, which at $n^{1/8}$ is less than $k = n^{1/4}$.
- The proof of Lemma 76 (p. 482) shows that at least half of the colorings assign different colors to nodes in X.
- So half of the negative examples have a clique in X and are accepted.

The Proof (continued)

- Recall the constants on p. 475: $k = n^{1/4}$, $\ell = n^{1/8}$, $p = n^{1/8} \log n$, $M = (p-1)^\ell \ell! < n^{(1/3)n^{1/8}}$ for large n.
- Suppose the final crude circuit is identically false.
  - By Lemma 81 (p. 494), each approximation step introduces at most $M^2 \binom{n-\ell-1}{k-\ell-1}$ false negatives.
  - There are $\binom{n}{k}$ positive examples.
  - The original crude circuit for $\text{CLIQUE}_{n,k}$ has at least

$$\frac{\binom{n}{k}}{M^2 \binom{n-\ell-1}{k-\ell-1}} \geq \frac{1}{M^2} \left(\frac{n-\ell}{k} \right)^\ell \geq n^{(1/12)n^{1/8}}$$

gates.

The Proof (concluded)

- Suppose the final crude circuit is not identically false.
  - Lemma 82 (p. 496) says that there are at least $(k-1)^n/2$ false positives.
  - By Lemma 80 (p. 494), each approximation step introduces at most $M^2 2^{-p}(k-1)^n$ false positives
  - The original crude circuit for $\text{CLIQUE}_{n,k}$ has at least

$$\frac{(k-1)^n/2}{M^2 2^{-p}(k-1)^n} = \frac{2^{p-1}}{M^2} \geq n^{(1/3)n^{1/8}}$$

gates.

Proving P $\neq$ NP?

- Razborov's theorem says that there is a monotone language in NP that has no polynomial monotone circuits.
- If we can prove that all monotone languages in P have polynomial monotone circuits, then P $\neq$ NP.
- But Razborov proved in 1985 that some monotone languages in P have no polynomial monotone circuits!

PSPACE and Games

- Given a boolean expression ϕ 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]

- We prove the result without imposing the CNF condition on ϕ.
- It is not hard to show that QSAT $\in$ PSPACE.
- Let L be a language decided by a polynomial-space TM M.
- There are at most 2^{n^k} configurations for some integer k given input x with $|x| = n$.
- Each configuration of M on input x can be coded as a bit vector of length n^k for some k.

[a] Stockmeyer, Meyer, 1973.

The Proof (continued)

- We need to write down a quantified boolean expression Ψ_i for expressing with free variables in set $A \cup B = \{a_1, \ldots, a_{n^k}, b_1, \ldots, b_{n^k}\}$.
- Ψ_i is true for some assignment to its free variables if and only if:
  - The true assignment for a_i's and b_i's encodes two configurations a and b.
  - There is a path from a to b in the configuration graph of length at most 2^i.

The Proof (continued)

- "$x \in L$" is $\Psi_{n^k}(A, B)$, where:
  - A is the truth assignment encoding the initial configuration.
  - B is the truth assignment encoding the accepting configuration.
- For $i = 0$, $\Psi_0(A, B)$ states that either $a_i = b_i$ for all i or configuration B follows from A in one step.
- This can be done in polynomial space.

The Proof (concluded)

- Inductively, suppose $\Psi_i(A, B)$ is available.
- $\Psi_{i+1}(A, B) \equiv \exists Z[\Psi_i(A, Z) \land \Psi_i(Z, B)]$ leads to exponentially large expressions.
- We need a way to use only one copy of Ψ_i.
- Here is how:

$$\Psi_{i+1}(A, B) \equiv \exists Z \forall X \forall Y$$
$$\{[(X = A \land Y = Z) \lor (X = Z \land Y = B)] \Rightarrow \Psi_i(X, Y)\}.$$

IP and PSPACE

- Shamir in 1990 proved that IP equals PSPACE.
- We will use a similar idea to prove that coNP $\subseteq$ IP.

Interactive Proof for Boolean Unsatisfiability

- A 3SAT formula is a conjunction of disjunctions of at most three literals.
- We shall present an interactive proof for boolean unsatisfiability.
- In other words, given an 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 $\rightarrow 0$
  - $x_i \rightarrow x_i$
  - $\bar{x_i} \rightarrow 1 - x_i$
  - $\vee \rightarrow +$
  - $\wedge \rightarrow \times$
  - $\phi(x_1, x_2, \ldots, x_n) \rightarrow \Phi(x_1, x_2, \ldots, x_n)$

The Arithmetic Version

- A boolean formula is transformed into a multivariate polynomial Φ.
- It is easy to verify that ϕ is unsatisfiable if and only if

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

Choosing the Field

- Suppose ϕ has m clauses of length three each.
- Then $\Phi \leq 3^m$.
- Because there are at most 2^n truth assignments,

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

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

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

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

Binding the Prover

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

The Algorithm

Peggy and Victor both arithmetize ϕ to obtain Φ;
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 to 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 $P_i^*(0) + P_i^*(1) \neq v_{i-1} \bmod q$ 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 $\Phi(r_1, r_2, \ldots, r_n) = v_n \bmod q$;

Remarks

- The following invariant is maintained by the algorithm:

$$P_i^*(0) + P_i^*(1) = P_{i-1}^*(r_{i-1}) \bmod q. \qquad (7)$$

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

Completeness

- Suppose ϕ is unsatisfiable.
- For $i \geq 1$,

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

Completeness (concluded)

- In particular at $i = 1$, because ϕ is unsatisfiable, we have
  $$P_1^*(0) + P_1^*(1) = \sum_{x_1=0,1} \cdots \sum_{x_n=0,1} \Phi(x_1, \ldots, x_n) = v_0 = 0 \bmod q.$$
- 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 ϕ.

Soundness

- Suppose ϕ is not unsatisfiable.
- An honest prover following the protocol will fail after sending $P_1^*(z)$.
- We will show that if the prover 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 as well.
- In the last round, her dishonesty is revealed.

Soundness (continued)

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

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

  (see Step 8 of the algorithm on p. 511).
- And so on to the end, except that the prover has no control over Step 12.

A Key Claim

Theorem 83 *If* $P_i^*(0) + P_i^*(1) \neq v_{i-1} \bmod q$*, then either the verifier rejects in the* i*th round, or* $P_i^*(r_i) \neq v_i \bmod q$ *with probability at least* $1 - (m/q)$*, where the probability is taken over the verifier's choices of* r_i*.*

The Proof of Theorem 83 (continued)

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

$$P_i(0) + P_i(1) = P_i^*(0) + P_i^*(1) \neq v_{i-1} \bmod q,$$

and the verifier rejects immediately.

- Suppose that the prover sends a $P_i(z)$ different from $P_i^*(z)$.
- If $P_i(z)$ does not pass the verifier's test $P_i(r_i) = v_i \bmod q$, then the verifier rejects.

The Proof of Theorem 83 (concluded)

- Assume $P_i(z)$ passes the test $P_i(r_i) = v_i \bmod q$.
- Because $P_i(z)$ and $P_i^*(z)$ are of degree at most m, there are at most m choices of $r_i \in Z_q$ such that

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

Soundness (continued)

- Suppose the verifier does not reject in any of the n rounds and exits the loop.
- As ϕ is not unsatisfiable,

$$P_1^*(0) + P_1^*(1) \neq v_0 \bmod q.$$

- By Theorem 83 (p. 517) and the fact that the verifier does not reject, we have $P_1^*(r_1) \neq v_1 \bmod q$ with probability at least $1 - (m/q)$.
- Now by (7),

$$P_1^*(r_1) = P_2^*(0) + P_2^*(1) \neq v_1 \bmod q.$$

Soundness (concluded)

- Iterating on this procedure, we eventually arrive at

$$P_n^*(r_n) \neq v_n \bmod q$$

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

- As $P_n^*(r_n) = \Phi(r_1, r_2, \ldots, r_n)$, the verifier's last test fails and he rejects.

- Altogether, the verifier fails with probability at least

$$(1 - m/q)^n > 1 - (nm/q) > 2/3$$

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

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.

Example (continued)

- The table below is an execution of the algorithm in Z_{73} *when the prover 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 ϕ early when $i = 1$.

Example (continued)

- Suppose Peggy does not follow the protocol.
- In order to deceive Victor, she comes up with fake polynomials $P_i(z)$'s from beginning to end.
- 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	10	61
2	$10z^2 + 9z + 21$	61	yes	4	71
3	$z^2 + 2z + 34$	71	yes	r_3	$P_3(r_3)$

Example (concluded)

- Now, Victor checks if the Φ satisfies

$$\Phi(10, 4, r_3) = P_3(r_3) \bmod 73.$$

- It can be verified that the only choices of $r_3 \in \{ 0, 1, \ldots, 72 \}$ that can mislead Victor are 10 and 12.
- The probability of that happening is only $2/73$.

Example

- $(x_1 \lor x_2) \land (x_1 \lor \neg x_2) \land (\neg x_1 \lor x_2) \land (\neg x_1 \lor \neg x_2)$.
- 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.

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 ϕ as unsatisfiable.

Objections to the Soundness Proof?[a]

- Based on the steps required of a cheating prover on p. 516, why must we go through so many rounds (in fact, n rounds)?
- Why not just go directly to round n:
  - The verifier sends $r_1, r_2, \ldots, r_{n-1}$ to the prover.
  - The prover returns with a (claimed) $P_n^*(z)$.
  - The verifier accepts if and only if $\Phi(r_1, r_2, \ldots, r_{n-1}, r_n) = P_n^*(r_n) \bmod q$ for a random $r_n \in Z_q$.

---

[a] Contributed by Mr. Chen and Ms. Hong in the lecture on January 2, 2002.

Objections to the Soundness Proof? (continued)

- Let us analyze the proposed compressed version when ϕ is satisfiable.
- To succeed in foiling the verifier, the prover must find a polynomial $P_n(z)$ of degree m such that $\Phi(r_1, r_2, \ldots, r_{n-1}, z) = P_n(z) \bmod q$.
- But this she is able to do: Just give the verifier polynomial $\Phi(r_1, r_2, \ldots, r_{n-1}, z)$!
- What happened?

Objections to the Soundness Proof? (concluded)

- You need the intermediate rounds to "tie" the prover up with a chain of claims.
- In the original algorithm on p. 511, for example, $P_n(z)$ is bound by the equality $P_n(0) + P_n(1) = v_{n-1} \bmod q$ 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 $P_{n-1}(0) + P_{n-1}(1) = v_{n-2} \bmod q$, and so on.

Finis

