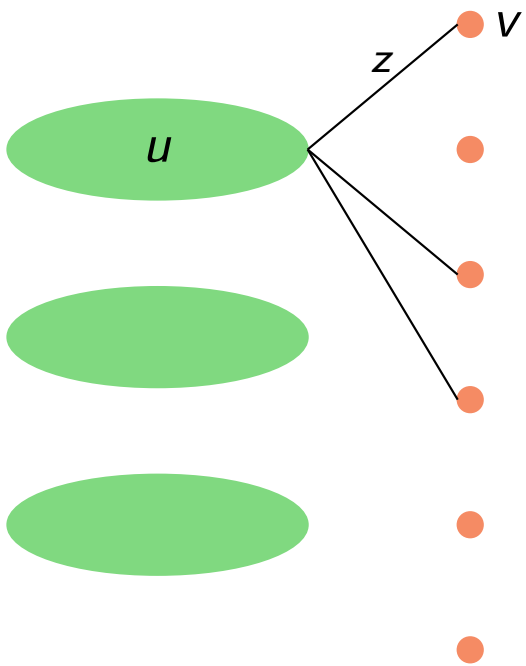
Towards an Optimal Query Efficient PCP?

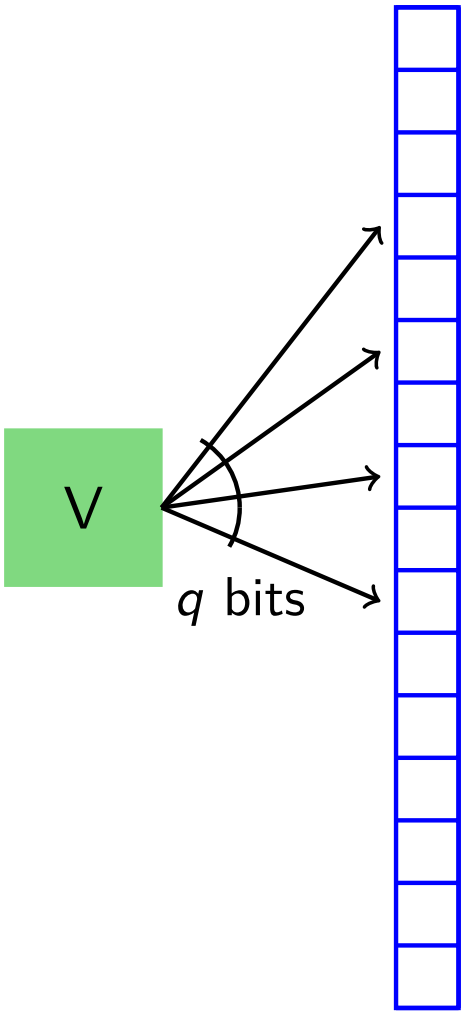


Subhash Khot
NYU and U. Chicago

Muli Safra
Tel Aviv University

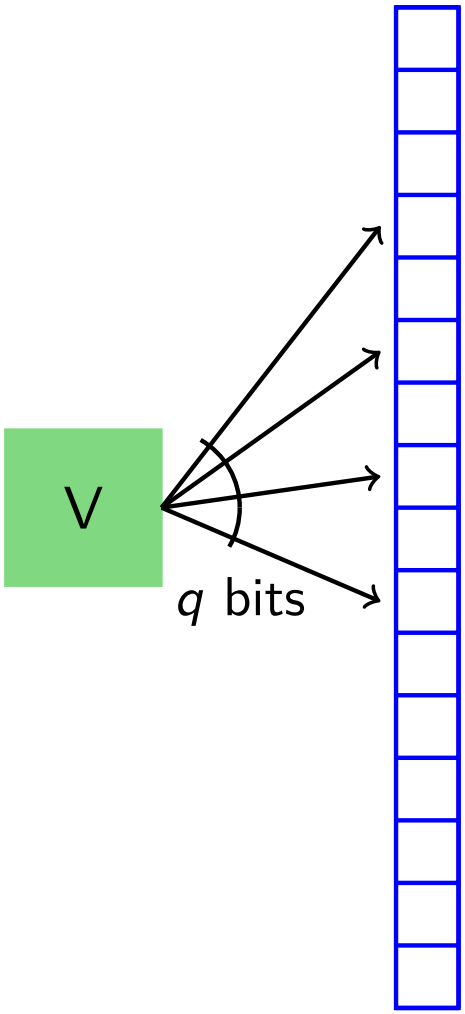
Madhur Tulsiani
TTI Chicago

Probabilistically Checkable Proofs (PCPs)



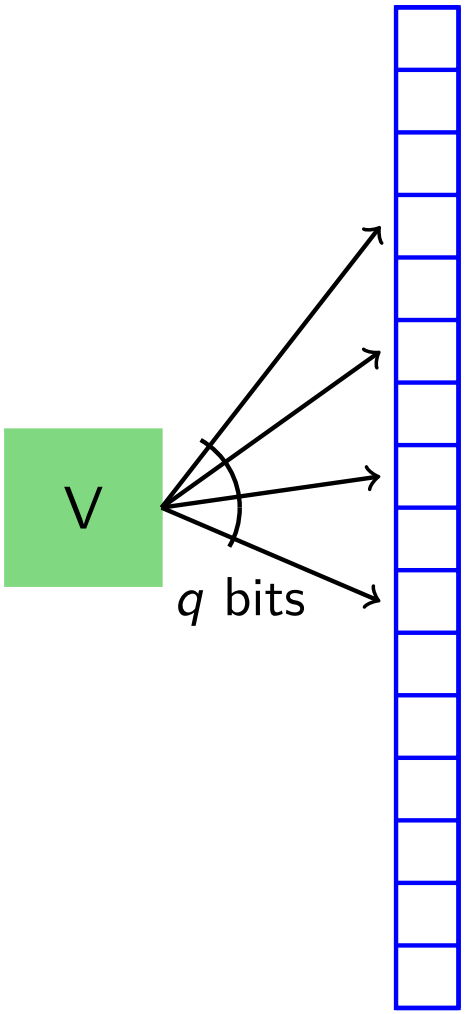
- Way of checking proof of satisfiability.

Probabilistically Checkable Proofs (PCPs)



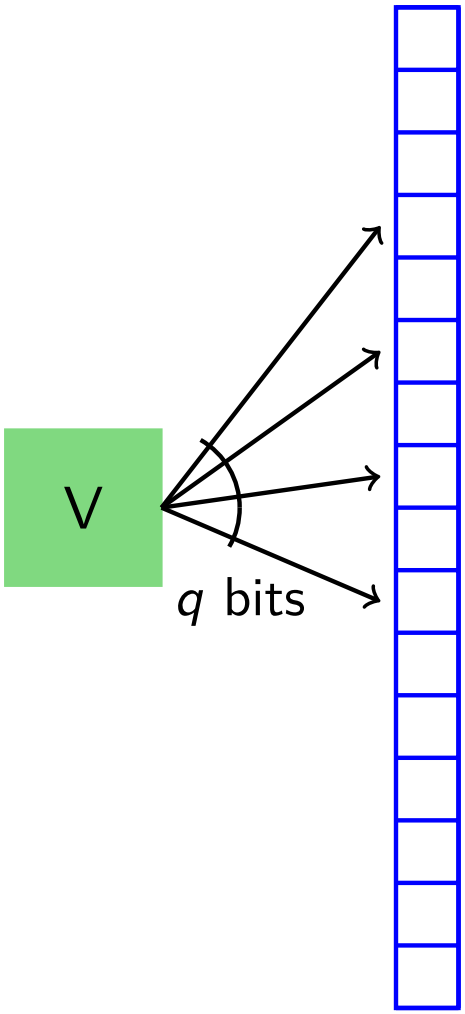
- Way of checking proof of satisfiability.
- SAT formula Ψ is satisfiable
  $\Longrightarrow$ Accept with probability ≈ 1
- Ψ is unsatisfiable
  $\Longrightarrow$ Accept with probability $\leq s$

Probabilistically Checkable Proofs (PCPs)

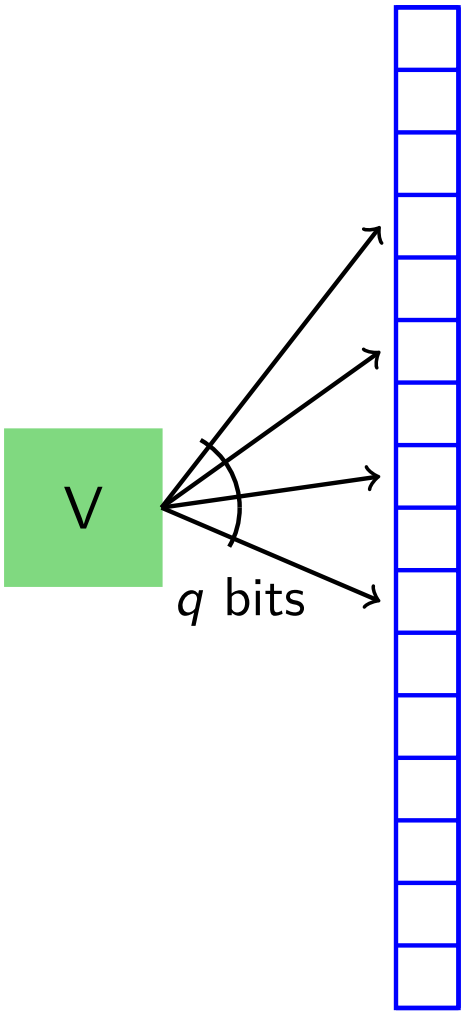


- Way of checking proof of satisfiability.
- SAT formula Ψ is satisfiable
  $\Longrightarrow$ Accept with probability ≈ 1
- Ψ is unsatisfiable
  $\Longrightarrow$ Accept with probability $\leq s$
- Interested in tradeoff of q vs s

PCPs $\equiv$ Constraint Satisfaction

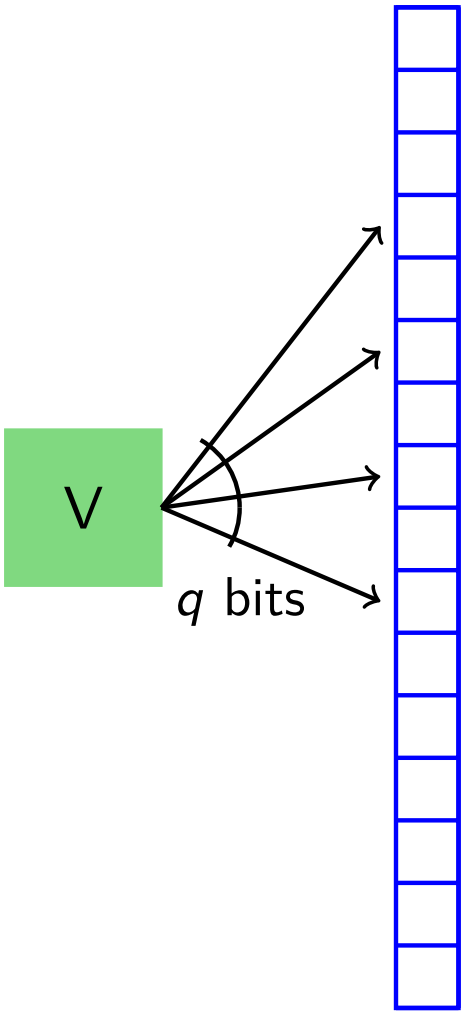


PCPs $\equiv$ Constraint Satisfaction



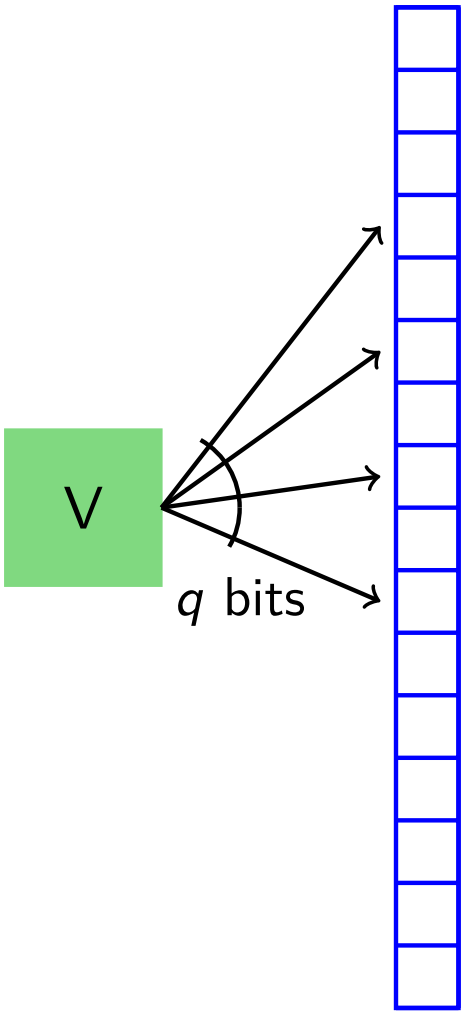
- Bits of proof $\equiv$ variables
  Verifier queries $\equiv$ constraints on q-tuples

PCPs $\equiv$ Constraint Satisfaction



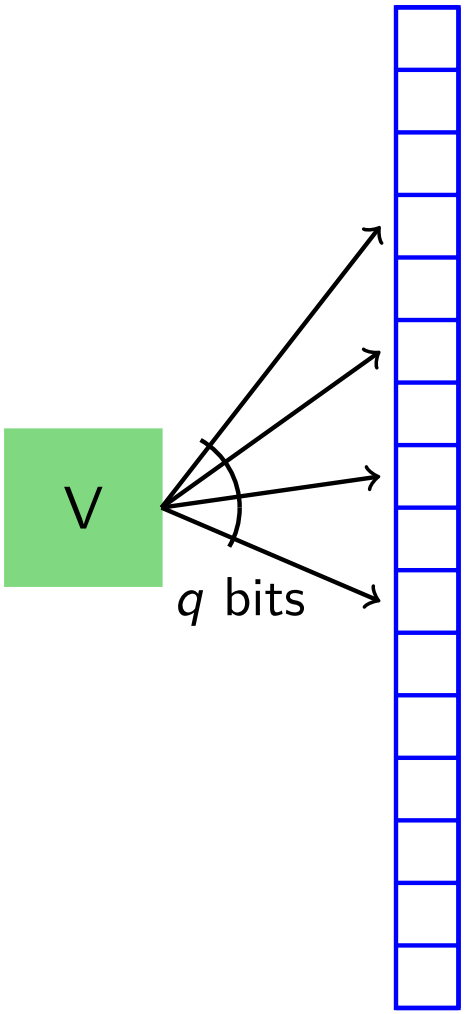
- Bits of proof $\equiv$ variables
  Verifier queries $\equiv$ constraints on q-tuples
- SAT formula Ψ is satisfiable
  $\Longrightarrow$ Fraction of constraints satisfied ≈ 1
- Ψ is unsatisfiable
  $\Longrightarrow$ Fraction of constraints satisfied $\leq s$

PCPs $\equiv$ Constraint Satisfaction



- Bits of proof $\equiv$ variables
  Verifier queries $\equiv$ constraints on q-tuples
- SAT formula Ψ is satisfiable
  $\Longrightarrow$ Fraction of constraints satisfied ≈ 1
- Ψ is unsatisfiable
  $\Longrightarrow$ Fraction of constraints satisfied $\leq s$
- Gives hardness of approximating within factor $1/s$.

PCPs $\equiv$ Constraint Satisfaction



- Bits of proof $\equiv$ variables
  Verifier queries $\equiv$ constraints on q-tuples
- SAT formula Ψ is satisfiable
  $\Longrightarrow$ Fraction of constraints satisfied ≈ 1
- Ψ is unsatisfiable
  $\Longrightarrow$ Fraction of constraints satisfied $\leq s$
- Gives hardness of approximating within factor $1/s$.
- Want to know best value of s using any kind of constraints on q variables.

Known Results

Known Results

[Håstad97]	$q = 3$	$s = 1/2$

Known Results

[Håstad97]	$q = 3$	$s = 1/2$
[ST00, HW01]	$q = 2k + k^2$	$s = 2^{-k^2} \approx \frac{2^{2\sqrt{q}}}{2^q}$
[EH05]	$q = k + \binom{k}{2}$	$s = 2^{-\binom{k}{2}} \approx \frac{2^{\sqrt{2q}}}{2^q}$

Known Results

[Håstad97]	$q = 3$	$s = 1/2$
[ST00, HW01]	$q = 2k + k^2$	$s = 2^{-k^2} \approx \frac{2^{2\sqrt{q}}}{2^q}$
[EH05]	$q = k + \binom{k}{2}$	$s = 2^{-\binom{k}{2}} \approx \frac{2^{\sqrt{2q}}}{2^q}$
[ST06] (UG-Hard)	$q = 2^k - 1$	$s = \frac{2^{k+1}}{2^{2^k}} \approx \frac{2q}{2^q}$

Known Results

[Håstad97]	$q = 3$	$s = 1/2$
[ST00, HW01]	$q = 2k + k^2$	$s = 2^{-k^2} \approx \frac{2^{2\sqrt{q}}}{2^q}$
[EH05]	$q = k + \binom{k}{2}$	$s = 2^{-\binom{k}{2}} \approx \frac{2^{\sqrt{2q}}}{2^q}$
[ST06] (UG-Hard)	$q = 2^k - 1$	$s = \frac{2^{k+1}}{2^{2^k}} \approx \frac{2q}{2^q}$
[AM09] (UG-Hard)*		$s \approx \frac{q + o(q)}{2^q}$

Known Results

[Håstad97]	$q = 3$	$s = 1/2$
[ST00, HW01]	$q = 2k + k^2$	$s = 2^{-k^2} \approx \frac{2^{2\sqrt{q}}}{2^q}$
[EH05]	$q = k + \binom{k}{2}$	$s = 2^{-\binom{k}{2}} \approx \frac{2^{\sqrt{2q}}}{2^q}$
[ST06] (UG-Hard)	$q = 2^k - 1$	$s = \frac{2^{k+1}}{2^{2^k}} \approx \frac{2q}{2^q}$
[AM09] (UG-Hard)*		$s \approx \frac{q + o(q)}{2^q}$
Here	$q = 2^3 - 1$	$s \leq \frac{1}{8} - \frac{1}{320}$

Known Results

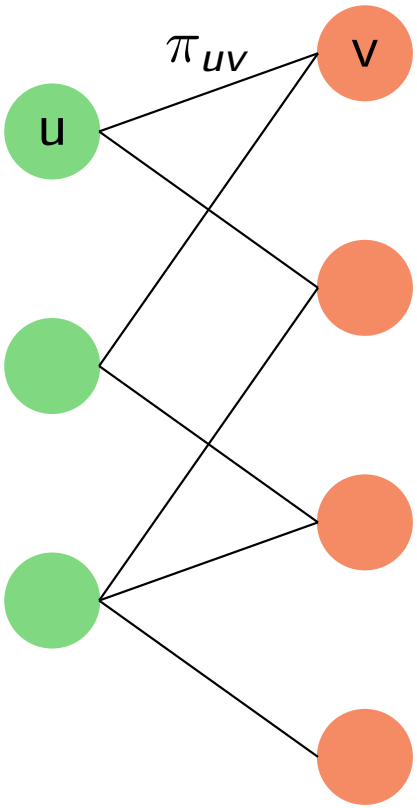
[Håstad97]	$q = 3$	$s = 1/2$
[ST00, HW01]	$q = 2k + k^2$	$s = 2^{-k^2} \approx \frac{2^{2\sqrt{q}}}{2^q}$
[EH05]	$q = k + \binom{k}{2}$	$s = 2^{-\binom{k}{2}} \approx \frac{2^{\sqrt{2q}}}{2^q}$
[ST06] (UG-Hard)	$q = 2^k - 1$	$s = \frac{2^{k+1}}{2^{2^k}} \approx \frac{2q}{2^q}$
[AM09] (UG-Hard)*		$s \approx \frac{q + o(q)}{2^q}$
Here	$q = 2^3 - 1$	$s \leq \frac{1}{8} - \frac{1}{320}$
[Chan12]	$q = 2^k - 1$	$s = \frac{2^{k+1}}{2^{2^k}} \approx \frac{2q}{2^q}$

Known Results

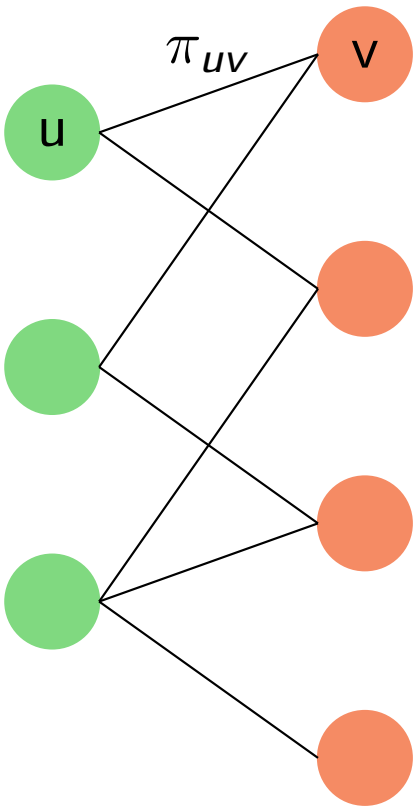
[Håstad97]	$q = 3$	$s = 1/2$
[ST00, HW01]	$q = 2k + k^2$	$s = 2^{-k^2} \approx \frac{2^{2\sqrt{q}}}{2^q}$
[EH05]	$q = k + \binom{k}{2}$	$s = 2^{-\binom{k}{2}} \approx \frac{2^{\sqrt{2q}}}{2^q}$
[ST06] (UG-Hard)	$q = 2^k - 1$	$s = \frac{2^{k+1}}{2^{2^k}} \approx \frac{2q}{2^q}$
[AM09] (UG-Hard)*		$s \approx \frac{q + o(q)}{2^q}$
Here	$q = 2^3 - 1$	$s \leq \frac{1}{8} - \frac{1}{320}$
[Chan12]	$q = 2^k - 1$	$s = \frac{2^{k+1}}{2^{2^k}} \approx \frac{2q}{2^q}$

- Interesting because of new outer PCP.

Outer PCPs: Two-Prover Games

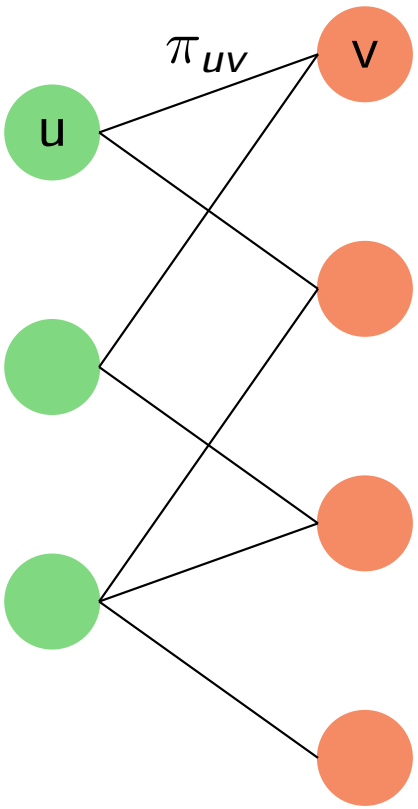


Outer PCPs: Two-Prover Games



- Verifier picks random edge (u, v). Sends u to one prover, v to other.
- Provers give $L(u) \in \Sigma_1$, $L(v) \in \Sigma_2$.

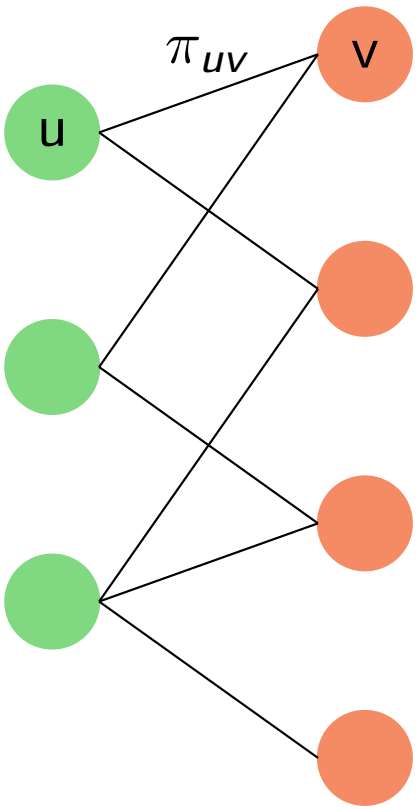
Outer PCPs: Two-Prover Games



- Verifier picks random edge (u, v). Sends u to one prover, v to other.
- Provers give $L(u) \in \Sigma_1$, $L(v) \in \Sigma_2$.
- Constraint given by projection $\pi_{uv} : \Sigma_1 \to \Sigma_2$. Accept if

$$\pi_{uv}(L(u)) \;=\; L(v)$$

Outer PCPs: Two-Prover Games

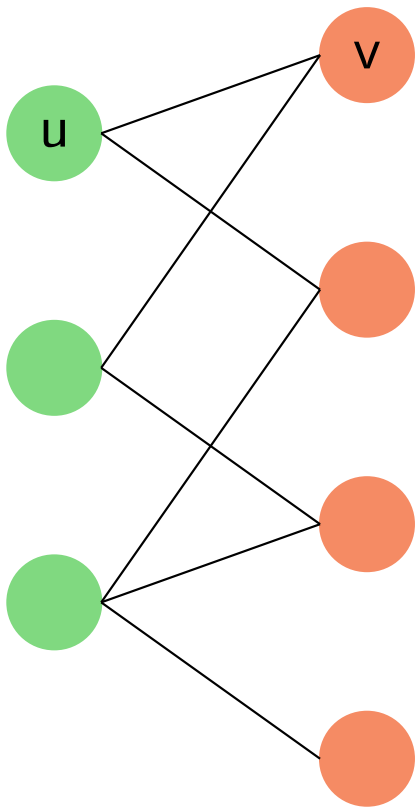


- Verifier picks random edge (u, v). Sends u to one prover, v to other.
- Provers give $L(u) \in \Sigma_1$, $L(v) \in \Sigma_2$.
- Constraint given by projection $\pi_{uv} : \Sigma_1 \to \Sigma_2$. Accept if

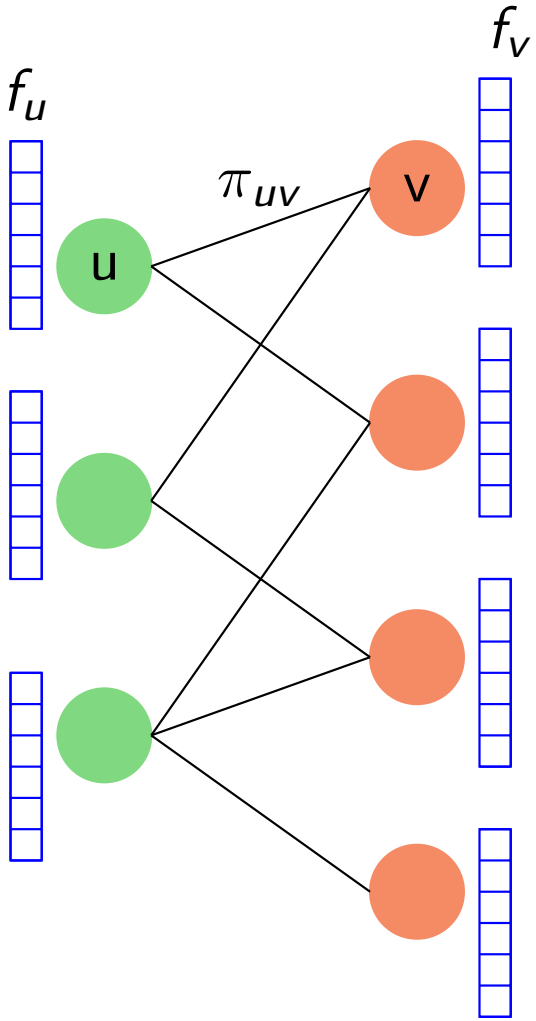
$$\pi_{uv}(L(u)) \;=\; L(v)$$

- Unique game when $\Sigma_1 = \Sigma_2$ and π_{uv}s are permutations.

Designing a PCP

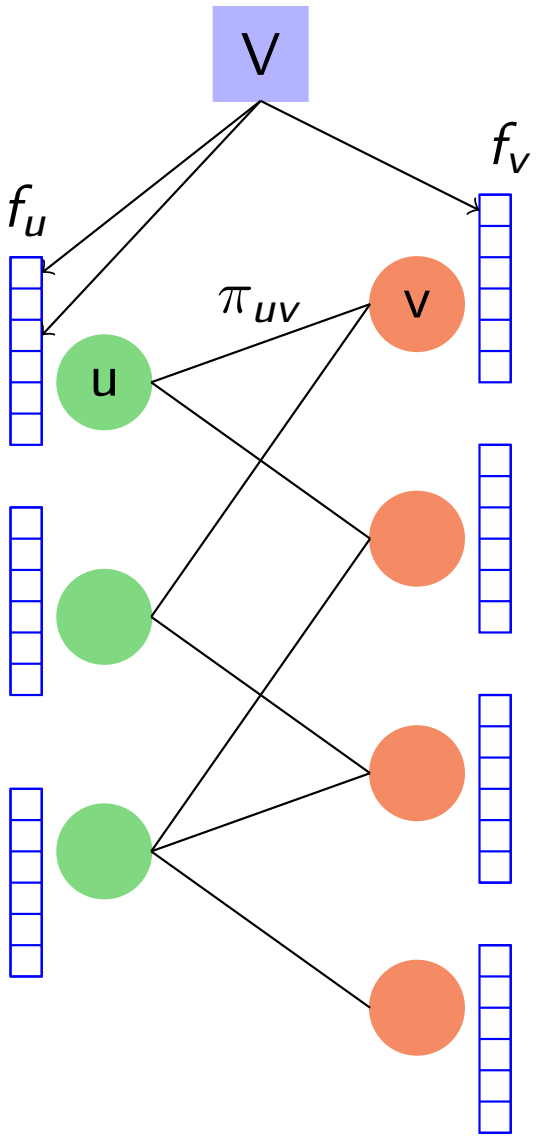


Designing a PCP



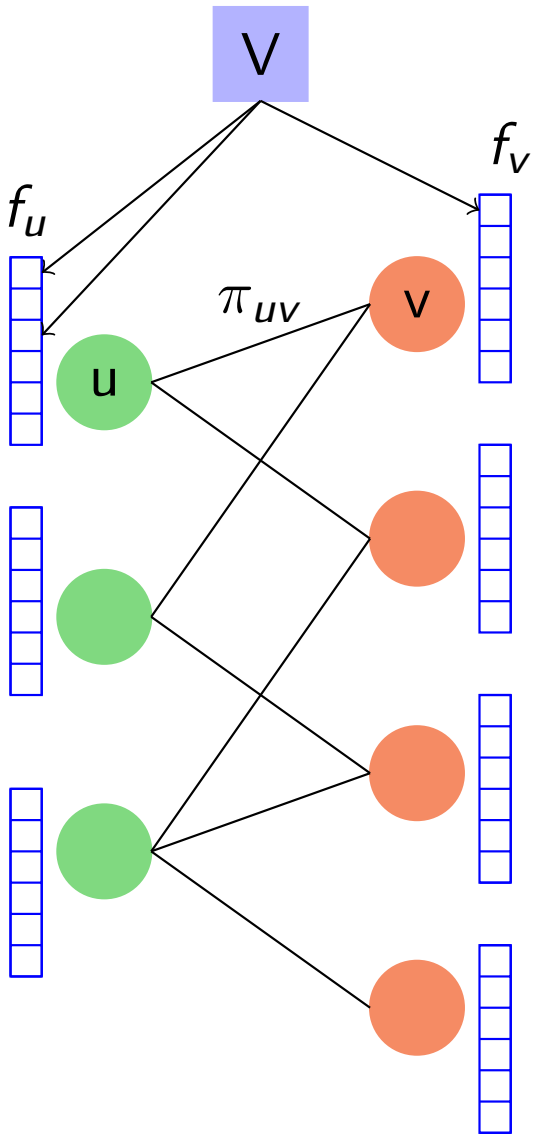
- Provers required to provide encodings of labels.

Designing a PCP



- Provers required to provide encodings of labels.
- Verifier makes q queries to encoded tables to check if
  - Encodings are valid.
  - Labelings satisfy outer PCP constraints.

Designing a PCP



- Provers required to provide encodings of labels.
- Verifier makes q queries to encoded tables to check if
  - Encodings are valid.
  - Labelings satisfy outer PCP constraints.
- Test for valid encodings (inner PCP) governs the type of constraints. e.g.

  $$f(x) + f(y) = f(x + y)$$

  in [Håstad97]

The inner PCP for [ST06]

- For $f : \{0,1\}^R \to \{0,1\}$, test as follows:
  - Pick $x_1, \ldots, x_k \in \{0,1\}^R$ at random.
  - For all $S \subseteq [k], |S| > 1$ test

$$f\left(\sum_{i \in S} x_i\right) = f\left(\sum_{i \in S} x_i\right)$$

The inner PCP for [ST06]

- For $f : \{0,1\}^R \to \{0,1\}$, test as follows:
  - Pick $x_1, \ldots, x_k \in \{0,1\}^R$ at random.
  - For all $S \subseteq [k], |S| > 1$ test

$$f\left(\sum_{i \in S} x_i\right) = f\left(\sum_{i \in S} x_i\right)$$

- Read f at $q = 2^k - 1$ places (one for each non-empty $S \subseteq [k]$)

The inner PCP for [ST06]

- For $f : \{0,1\}^R \to \{0,1\}$, test as follows:
  - Pick $x_1, \ldots, x_k \in \{0,1\}^R$ at random.
  - For all $S \subseteq [k], |S| > 1$ test

$$f\left(\sum_{i \in S} x_i\right) = f\left(\sum_{i \in S} x_i\right)$$

- Read f at $q = 2^k - 1$ places (one for each non-empty $S \subseteq [k]$)
- No known outer PCP (except Unique Games) which this could be combined.

The inner PCP for [ST06]

- For $f : \{0,1\}^R \to \{0,1\}$, test as follows:
  - Pick $x_1, \ldots, x_k \in \{0,1\}^R$ at random.
  - For all $S \subseteq [k], |S| > 1$ test

$$f\left(\sum_{i \in S} x_i\right) = f\left(\sum_{i \in S} x_i\right)$$

- Read f at $q = 2^k - 1$ places (one for each non-empty $S \subseteq [k]$)
- No known outer PCP (except Unique Games) which this could be combined.
- We give such an outer PCP.

The case of $k = 3$

- For all x_1, x_2, x_3, we have a constraint checking all of the following

$$\begin{aligned} f(x_1) + f(x_2) &= f(x_1 + x_2) \\ f(x_2) + f(x_3) &= f(x_2 + x_3) \\ f(x_3) + f(x_1) &= f(x_3 + x_1) \\ f(x_1) + f(x_2) + f(x_3) &= f(x_1 + x_2 + x_3) \end{aligned}$$

The case of $k = 3$

- For all x_1, x_2, x_3, we have a constraint checking all of the following

$$\begin{aligned} f(x_1) + f(x_2) &= f(x_1 + x_2) \\ f(x_2) + f(x_3) &= f(x_2 + x_3) \\ f(x_3) + f(x_1) &= f(x_3 + x_1) \\ f(x_1) + f(x_2) + f(x_3) &= f(x_1 + x_2 + x_3) \end{aligned}$$

- $q = 7$. $s = 1/8$ follows from [EH05]. [ST06] achieves $s = 1/16$ under UGC.

The case of $k = 3$

- For all x_1, x_2, x_3, we have a constraint checking all of the following

$$\begin{aligned} f(x_1) + f(x_2) &= f(x_1 + x_2) \\ f(x_2) + f(x_3) &= f(x_2 + x_3) \\ f(x_3) + f(x_1) &= f(x_3 + x_1) \\ f(x_1) + f(x_2) + f(x_3) &= f(x_1 + x_2 + x_3) \end{aligned}$$

- $q = 7$. $s = 1/8$ follows from [EH05]. [ST06] achieves $s = 1/16$ under UGC.

- Random quadratic f passes above test with probability $1/8$.

The case of $k = 3$

- For all x_1, x_2, x_3, we have a constraint checking all of the following

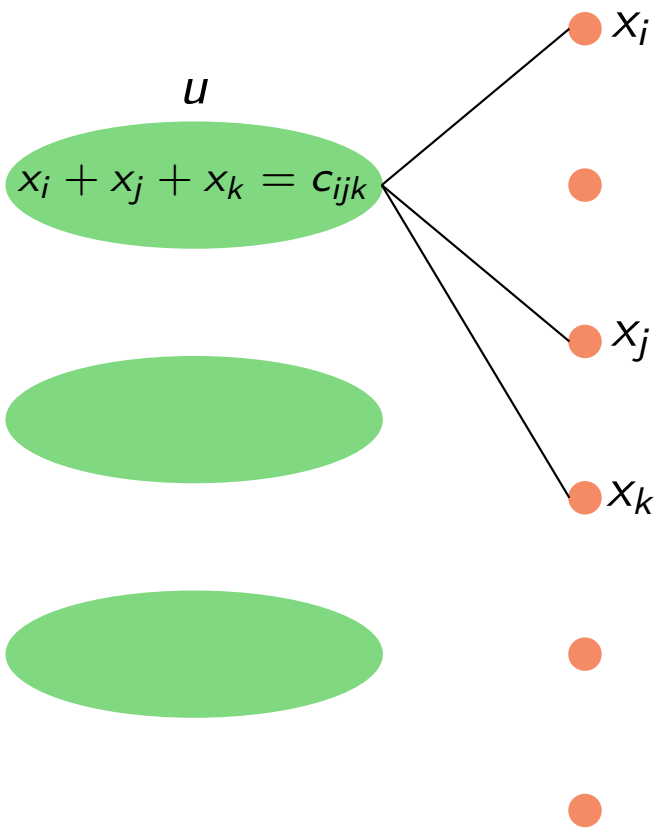
$$\begin{aligned} f(x_1) + f(x_2) &= f(x_1 + x_2) \\ f(x_2) + f(x_3) &= f(x_2 + x_3) \\ f(x_3) + f(x_1) &= f(x_3 + x_1) \\ f(x_1) + f(x_2) + f(x_3) &= f(x_1 + x_2 + x_3) \end{aligned}$$

- $q = 7$. $s = 1/8$ follows from [EH05]. [ST06] achieves $s = 1/16$ under UGC.

- Random quadratic f passes above test with probability $1/8$.

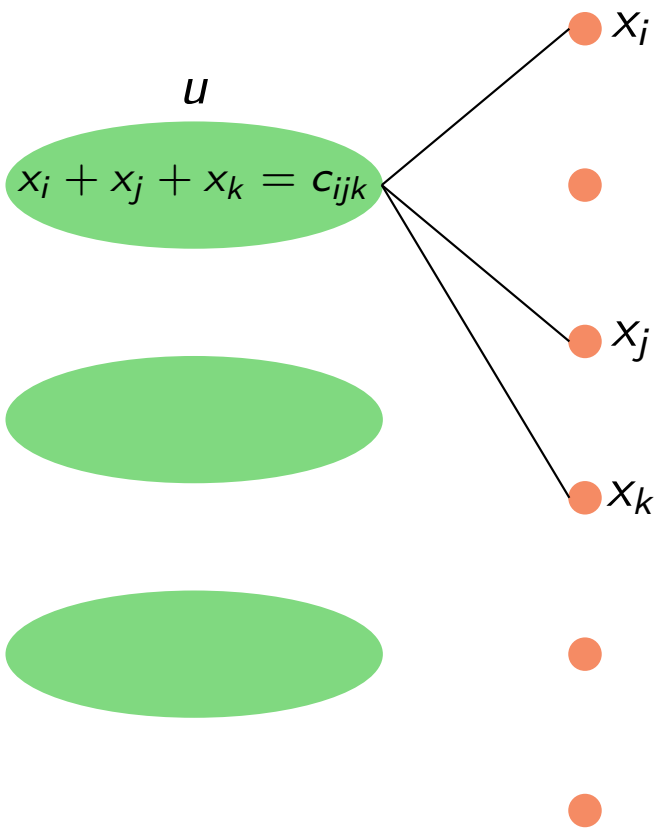
- Quadratic functions provide (somewhat) good encodings. Outer PCP needs to be robust against quadratic encodings.

Robustness against linear encodings (using [Håstad97])



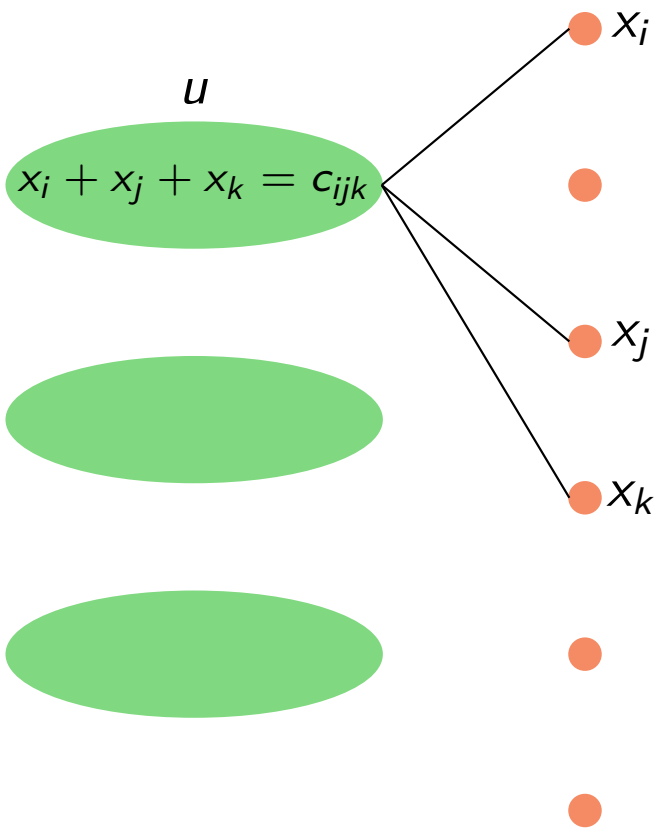
- Outer PCP from instance of 3-XOR.

Robustness against linear encodings (using [Håstad97])



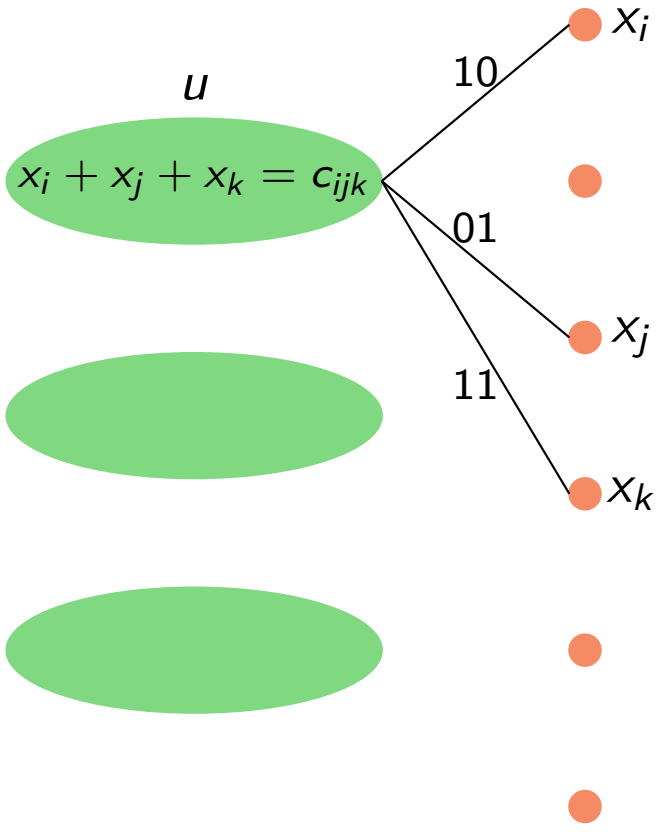
- Outer PCP from instance of 3-XOR.
- Labels on left are functions $f_u : \{0,1\}^2 \to \{0,1\}$. Labels on right are bits.

Robustness against linear encodings (using [Håstad97])



- Outer PCP from instance of 3-XOR.
- Labels on left are functions $f_u : \{0,1\}^2 \to \{0,1\}$. Labels on right are bits.
- Intended: $f_u(z_1, z_2) = z_1 \cdot L(x_1) + z_2 \cdot L(x_2)$.

Robustness against linear encodings (using [Håstad97])

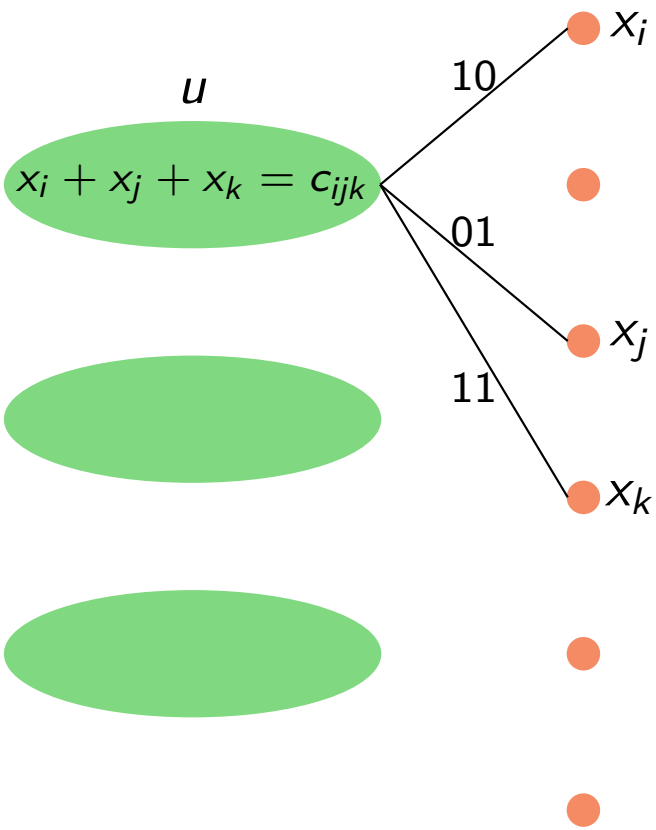


- Outer PCP from instance of 3-XOR.
- Labels on left are functions $f_u : \{0,1\}^2 \to \{0,1\}$. Labels on right are bits.
- Intended: $f_u(z_1, z_2) = z_1 \cdot L(x_1) + z_2 \cdot L(x_2)$.
- Constraints for different inputs to f_u.

$$\begin{aligned} f_u(1,0) &= L(x_i) \\ f_u(0,1) &= L(x_j) \\ f_u(1,1) &= L(x_k) + c_{ijk} \end{aligned}$$

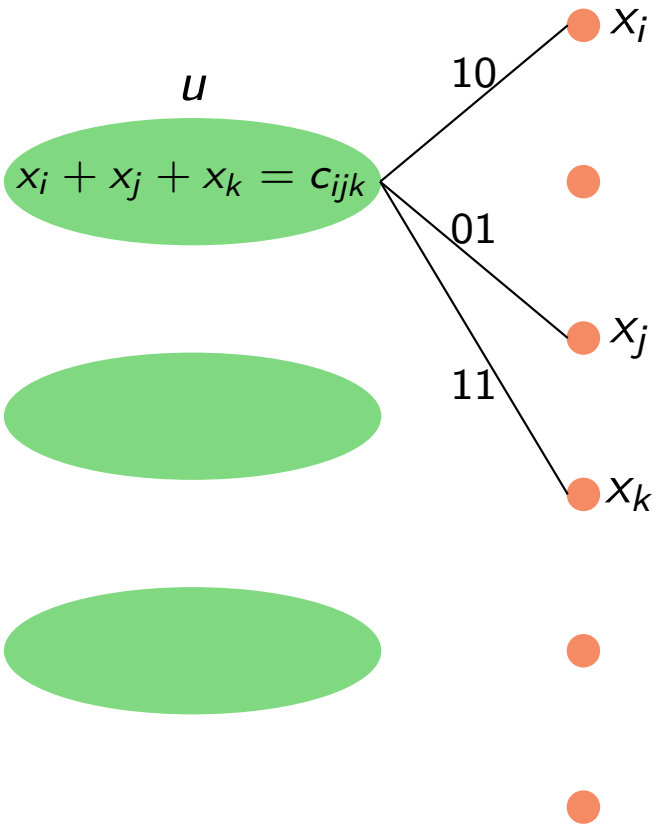
(must have $L(x_i) + L(x_i) = L(x_k) + c_{ijk}$).

Robustness against linear encodings



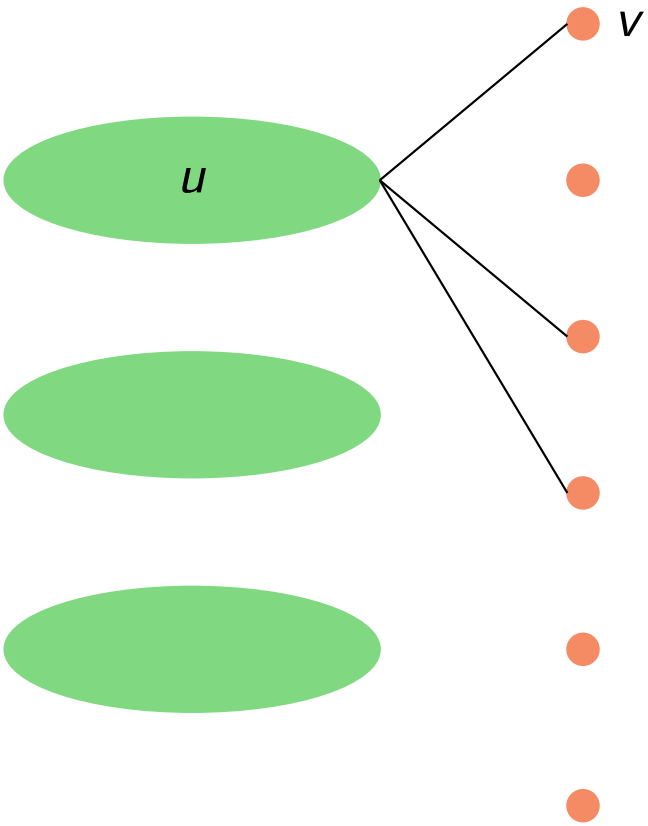
- [Håstad97]: Hard to distinguish:
  - Almost all XOR constraints can be satisfied.
  - At most $1/2$ of XOR constraints can be satisfied.

Robustness against linear encodings



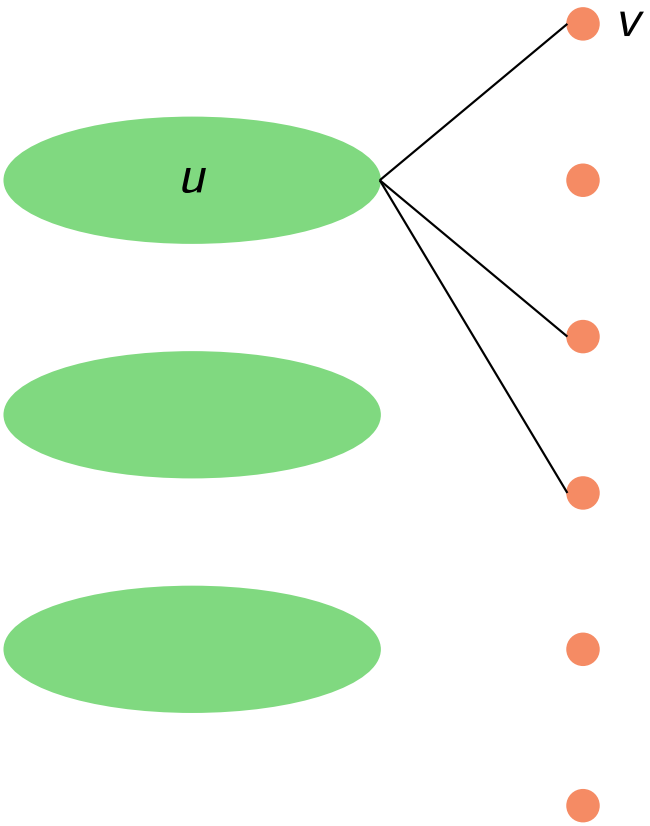
- [Håstad97]: Hard to distinguish:
  - Almost all XOR constraints can be satisfied.
  - At most 1/2 of XOR constraints can be satisfied.
- Gives hardness of distinguishing between the following cases:
  - $\exists$ linear functions $\{f_u\}$ satisfying ≈ 1 fraction of constraints.
  - $\forall$ sets of linear functions $\{f_u\}$, at most 5/6 fraction of constraints are satisfied.

Robustness against degree-d polynomials



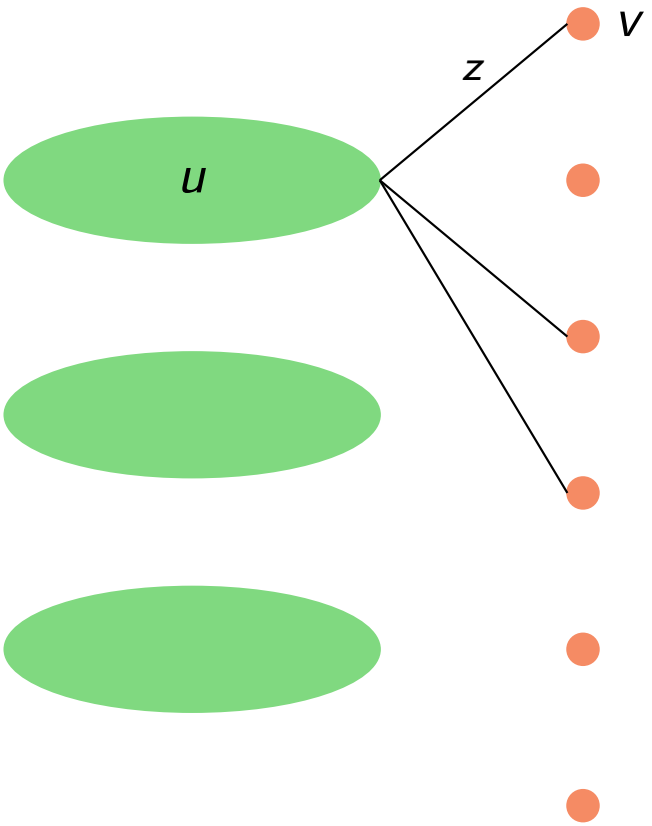
- We construct a variant of Håstad's PCP.

Robustness against degree-d polynomials



- We construct a variant of Håstad's PCP.
- Labels on left are functions $f_u : \{0,1\}^{d+1} \to \{0,1\}$. Labels on right are bits.

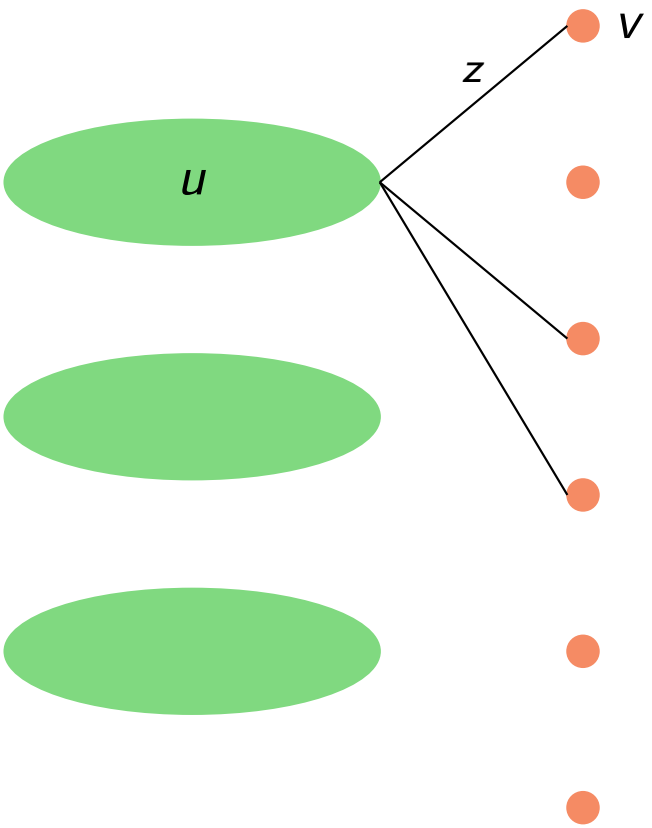
Robustness against degree-d polynomials



- We construct a variant of Håstad's PCP.
- Labels on left are functions $f_u : \{0,1\}^{d+1} \to \{0,1\}$. Labels on right are bits.
- Constraint on edge (u, v) labeled by an input z to f_u. Constraints of the form

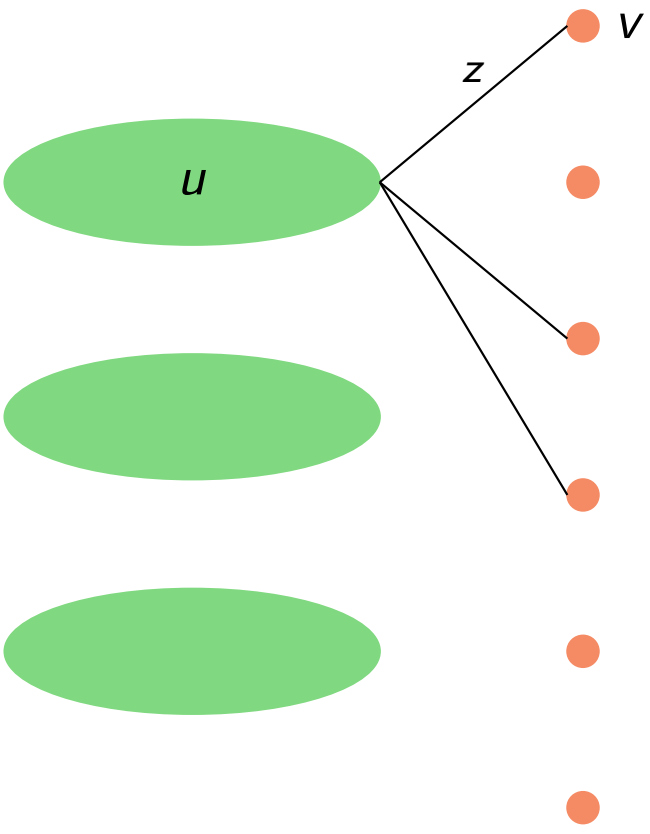
$$f_u(z) \;=\; L(v) + c_{uv,z}.$$

Linear vs degree-d labeling



- Prove hardness of distinguishing between the following cases:
  - $\exists$ linear functions $\{f_u\}$ satisfying ≈ 1 fraction of constraints.
  - $\forall$ sets of degree-d polynomial functions $\{f_u\}$, at most $1-\gamma$ fraction of constraints are satisfied ($\gamma = 2^{-O(d)}$).

Linear vs degree-d labeling



- Prove hardness of distinguishing between the following cases:
  - $\exists$ linear functions $\{f_u\}$ satisfying ≈ 1 fraction of constraints.
  - $\forall$ sets of degree-d polynomial functions $\{f_u\}$, at most $1-\gamma$ fraction of constraints are satisfied ($\gamma = 2^{-O(d)}$).
- Strengthen via parallel repetition to combine with inner PCP.

To do

- Other applications of the outer PCP.

To do

- Other applications of the outer PCP.
- Lowest value of s with perfect completeness (when verifier accepts a valid proof with probability exactly 1). Best is still $s = 2^{4\sqrt{q}}/2^q$ from [HK05].

To do

- Other applications of the outer PCP.
- Lowest value of s with perfect completeness (when verifier accepts a valid proof with probability exactly 1). Best is still $s = 2^{4\sqrt{q}}/2^q$ from [HK05].
- Even a version of the outer PCP here with perfect completeness would be interesting. Cannot have linear functions in the first case.

Thank You

Questions?