NEEXP in MIP*

Hyuntaek Shin
Dept. of Data Science

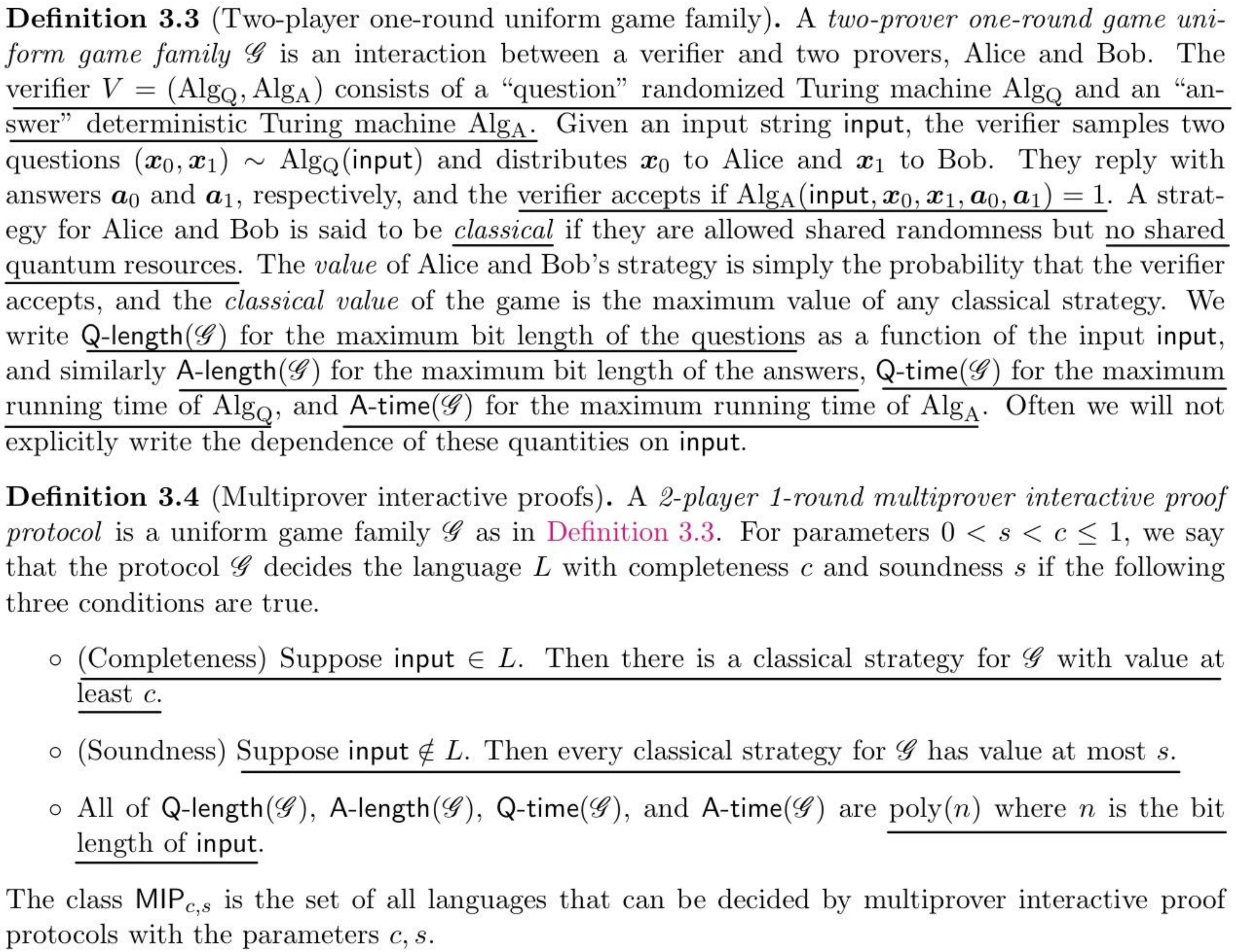
QISCA Quantum Complexity Theory Study
2025. 11. 01

Contents

- 0. Introduction and Motivation
- 1. SvP and Classical PCP Theorem
- 2. Question Reduction
- 3. Answer Reduction

0. Introduction and Motivation

2-player 1-round MIP protocol and MIP



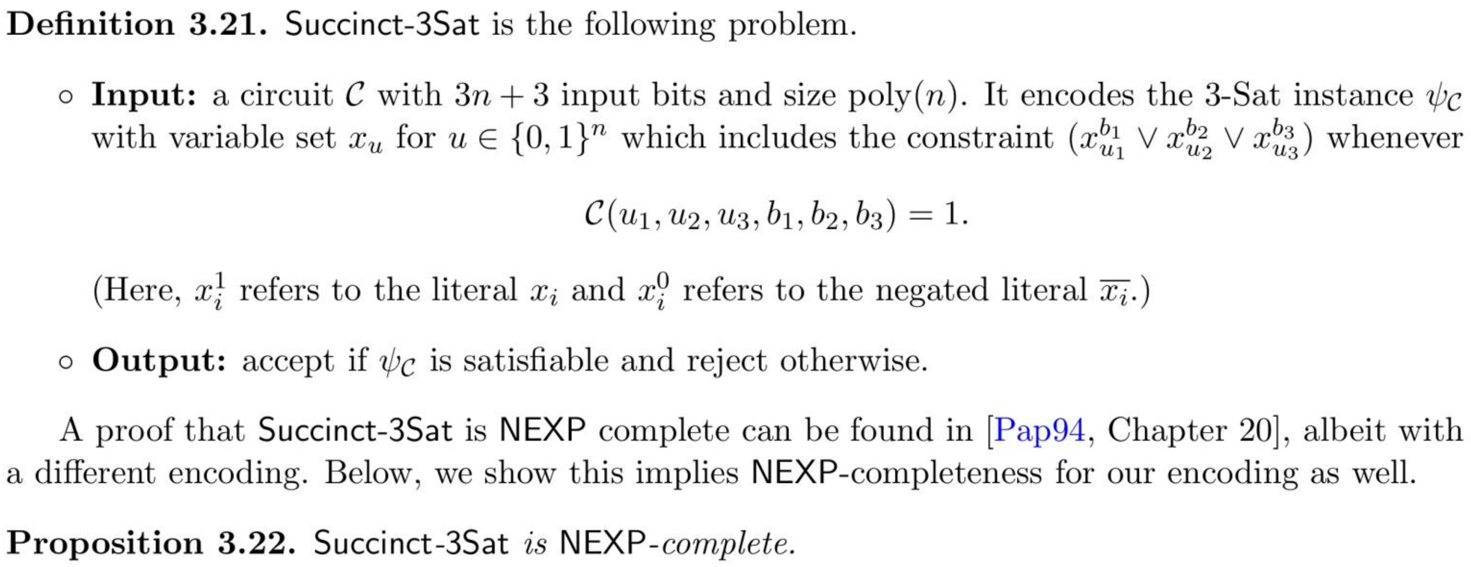
Definition 3.3 (Two-player one-round uniform game family)**.** A *two-prover one-round game uniform game family* $\mathscr{G}$ is an interaction between a verifier and two provers, Alice and Bob. The verifier $V = (\mathrm{Alg}_\mathrm{Q}, \mathrm{Alg}_\mathrm{A})$ consists of a "question" randomized Turing machine Alg_Q and an "answer" deterministic Turing machine Alg_A. Given an input string input, the verifier samples two questions $(\boldsymbol{x}_0, \boldsymbol{x}_1) \sim \mathrm{Alg}_\mathrm{Q}(\mathsf{input})$ and distributes $\boldsymbol{x}_0$ to Alice and $\boldsymbol{x}_1$ to Bob. They reply with answers $\boldsymbol{a}_0$ and $\boldsymbol{a}_1$, respectively, and the verifier accepts if $\mathrm{Alg}_\mathrm{A}(\mathsf{input}, \boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{a}_0, \boldsymbol{a}_1) = 1$. A strategy for Alice and Bob is said to be *classical* if they are allowed shared randomness but no shared quantum resources. The *value* of Alice and Bob's strategy is simply the probability that the verifier accepts, and the *classical value* of the game is the maximum value of any classical strategy. We write $\mathsf{Q\text{-}length}(\mathscr{G})$ for the maximum bit length of the questions as a function of the input input, and similarly $\mathsf{A\text{-}length}(\mathscr{G})$ for the maximum bit length of the answers, $\mathsf{Q\text{-}time}(\mathscr{G})$ for the maximum running time of Alg_Q, and $\mathsf{A\text{-}time}(\mathscr{G})$ for the maximum running time of Alg_A. Often we will not explicitly write the dependence of these quantities on input.

Definition 3.4 (Multiprover interactive proofs)**.** A *2-player 1-round multiprover interactive proof protocol* is a uniform game family $\mathscr{G}$ as in Definition 3.3. For parameters $0 < s < c \leq 1$, we say that the protocol $\mathscr{G}$ decides the language L with completeness c and soundness s if the following three conditions are true.

- (Completeness) Suppose $\mathsf{input} \in L$. Then there is a classical strategy for $\mathscr{G}$ with value at least c.
- (Soundness) Suppose $\mathsf{input} \notin L$. Then every classical strategy for $\mathscr{G}$ has value at most s.
- All of $\mathsf{Q\text{-}length}(\mathscr{G})$, $\mathsf{A\text{-}length}(\mathscr{G})$, $\mathsf{Q\text{-}time}(\mathscr{G})$, and $\mathsf{A\text{-}time}(\mathscr{G})$ are $\mathrm{poly}(n)$ where n is the bit length of input.

The class $\mathsf{MIP}_{c,s}$ is the set of all languages that can be decided by multiprover interactive proof protocols with the parameters c, s.

Succinct-3SAT and S-S-3SAT

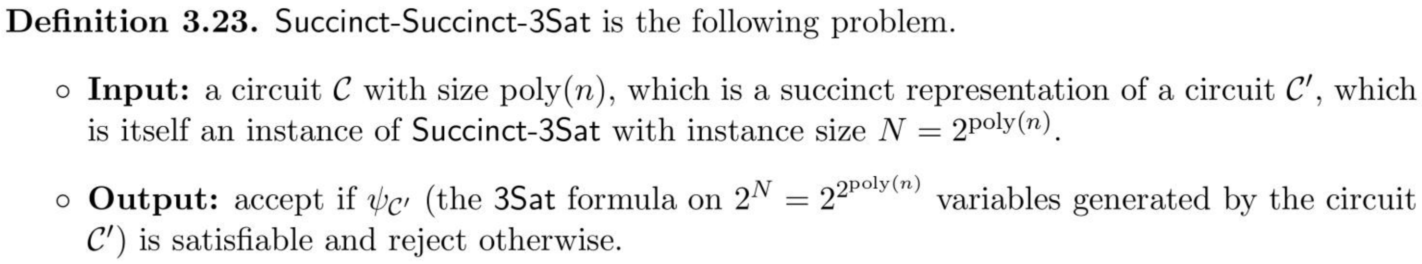


Definition 3.21. Succinct-3Sat is the following problem.

- **Input:** a circuit $\mathcal{C}$ with $3n + 3$ input bits and size $\mathrm{poly}(n)$. It encodes the 3-Sat instance $\psi_{\mathcal{C}}$ with variable set x_u for $u \in \{0,1\}^n$ which includes the constraint $(x_{u_1}^{b_1} \vee x_{u_2}^{b_2} \vee x_{u_3}^{b_3})$ whenever
$$\mathcal{C}(u_1, u_2, u_3, b_1, b_2, b_3) = 1.$$
(Here, x_i^1 refers to the literal x_i and x_i^0 refers to the negated literal $\overline{x_i}$.)
- **Output:** accept if $\psi_{\mathcal{C}}$ is satisfiable and reject otherwise.

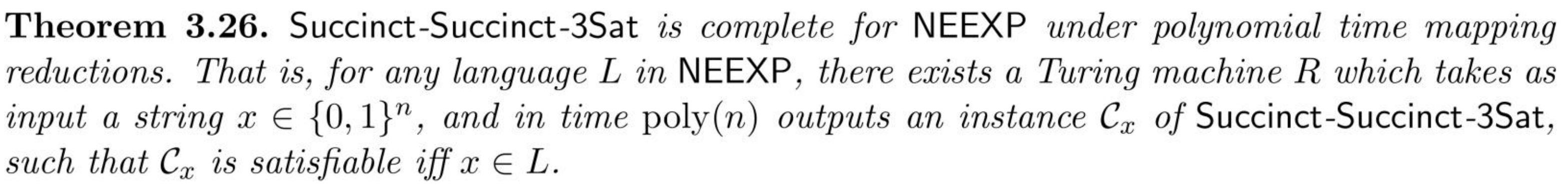
A proof that Succinct-3Sat is NEXP complete can be found in [Pap94, Chapter 20], albeit with a different encoding. Below, we show this implies NEXP-completeness for our encoding as well.

Proposition 3.22. Succinct-3Sat *is* NEXP-*complete.*



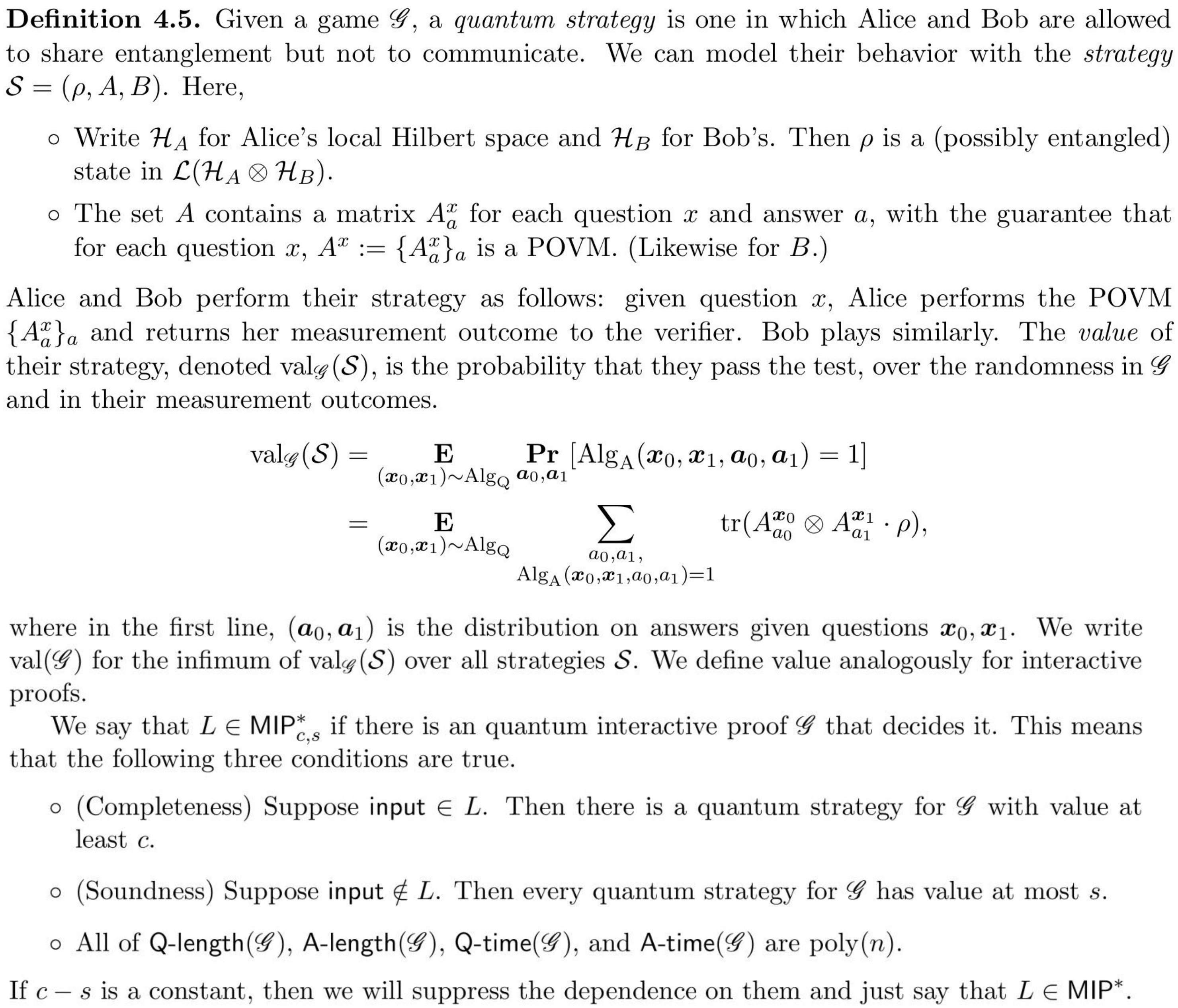
Definition 3.23. Succinct-Succinct-3Sat is the following problem.

- **Input:** a circuit $\mathcal{C}$ with size $\mathrm{poly}(n)$, which is a succinct representation of a circuit $\mathcal{C}'$, which is itself an instance of Succinct-3Sat with instance size $N = 2^{\mathrm{poly}(n)}$.
- **Output:** accept if $\psi_{\mathcal{C}'}$ (the 3Sat formula on $2^N = 2^{2^{\mathrm{poly}(n)}}$ variables generated by the circuit $\mathcal{C}'$) is satisfiable and reject otherwise.



Theorem 3.26. Succinct-Succinct-3Sat *is complete for* NEEXP *under polynomial time mapping reductions. That is, for any language L in* NEEXP, *there exists a Turing machine R which takes as input a string $x \in \{0,1\}^n$, and in time* $\mathrm{poly}(n)$ *outputs an instance $\mathcal{C}_x$ of* Succinct-Succinct-3Sat, *such that $\mathcal{C}_x$ is satisfiable iff $x \in L$.*

Quantum strategy and MIP*



Definition 4.5. Given a game $\mathscr{G}$, a *quantum strategy* is one in which Alice and Bob are allowed to share entanglement but not to communicate. We can model their behavior with the *strategy* $\mathcal{S} = (\rho, A, B)$. Here,

- Write $\mathcal{H}_A$ for Alice's local Hilbert space and $\mathcal{H}_B$ for Bob's. Then ρ is a (possibly entangled) state in $\mathcal{L}(\mathcal{H}_A \otimes \mathcal{H}_B)$.
- The set A contains a matrix A^x_a for each question x and answer a, with the guarantee that for each question x, $A^x := \{A^x_a\}_a$ is a POVM. (Likewise for B.)

Alice and Bob perform their strategy as follows: given question x, Alice performs the POVM $\{A^x_a\}_a$ and returns her measurement outcome to the verifier. Bob plays similarly. The *value* of their strategy, denoted $\mathrm{val}_{\mathscr{G}}(\mathcal{S})$, is the probability that they pass the test, over the randomness in $\mathscr{G}$ and in their measurement outcomes.

$$\begin{aligned}\mathrm{val}_{\mathscr{G}}(\mathcal{S}) &= \mathop{\mathbf{E}}_{(\boldsymbol{x}_0,\boldsymbol{x}_1)\sim \mathrm{Alg}_{\mathrm{Q}}} \ \mathop{\mathbf{Pr}}_{\boldsymbol{a}_0,\boldsymbol{a}_1} [\mathrm{Alg}_{\mathrm{A}}(\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{a}_0, \boldsymbol{a}_1) = 1] \\ &= \mathop{\mathbf{E}}_{(\boldsymbol{x}_0,\boldsymbol{x}_1)\sim \mathrm{Alg}_{\mathrm{Q}}} \sum_{\substack{a_0, a_1, \\ \mathrm{Alg}_{\mathrm{A}}(\boldsymbol{x}_0,\boldsymbol{x}_1,a_0,a_1)=1}} \mathrm{tr}(A^{\boldsymbol{x}_0}_{a_0} \otimes A^{\boldsymbol{x}_1}_{a_1} \cdot \rho),\end{aligned}$$

where in the first line, $(\boldsymbol{a}_0, \boldsymbol{a}_1)$ is the distribution on answers given questions $\boldsymbol{x}_0, \boldsymbol{x}_1$. We write $\mathrm{val}(\mathscr{G})$ for the infimum of $\mathrm{val}_{\mathscr{G}}(\mathcal{S})$ over all strategies $\mathcal{S}$. We define value analogously for interactive proofs.

We say that $L \in \mathsf{MIP}^*_{c,s}$ if there is an quantum interactive proof $\mathscr{G}$ that decides it. This means that the following three conditions are true.

- (Completeness) Suppose $\mathsf{input} \in L$. Then there is a quantum strategy for $\mathscr{G}$ with value at least c.
- (Soundness) Suppose $\mathsf{input} \notin L$. Then every quantum strategy for $\mathscr{G}$ has value at most s.
- All of $\mathsf{Q\text{-}length}(\mathscr{G})$, $\mathsf{A\text{-}length}(\mathscr{G})$, $\mathsf{Q\text{-}time}(\mathscr{G})$, and $\mathsf{A\text{-}time}(\mathscr{G})$ are $\mathrm{poly}(n)$.

If $c - s$ is a constant, then we will suppress the dependence on them and just say that $L \in \mathsf{MIP}^*$.

Motivation

Since S-S-3SAT problem is complete under NEEXP by polynomial time reduction,

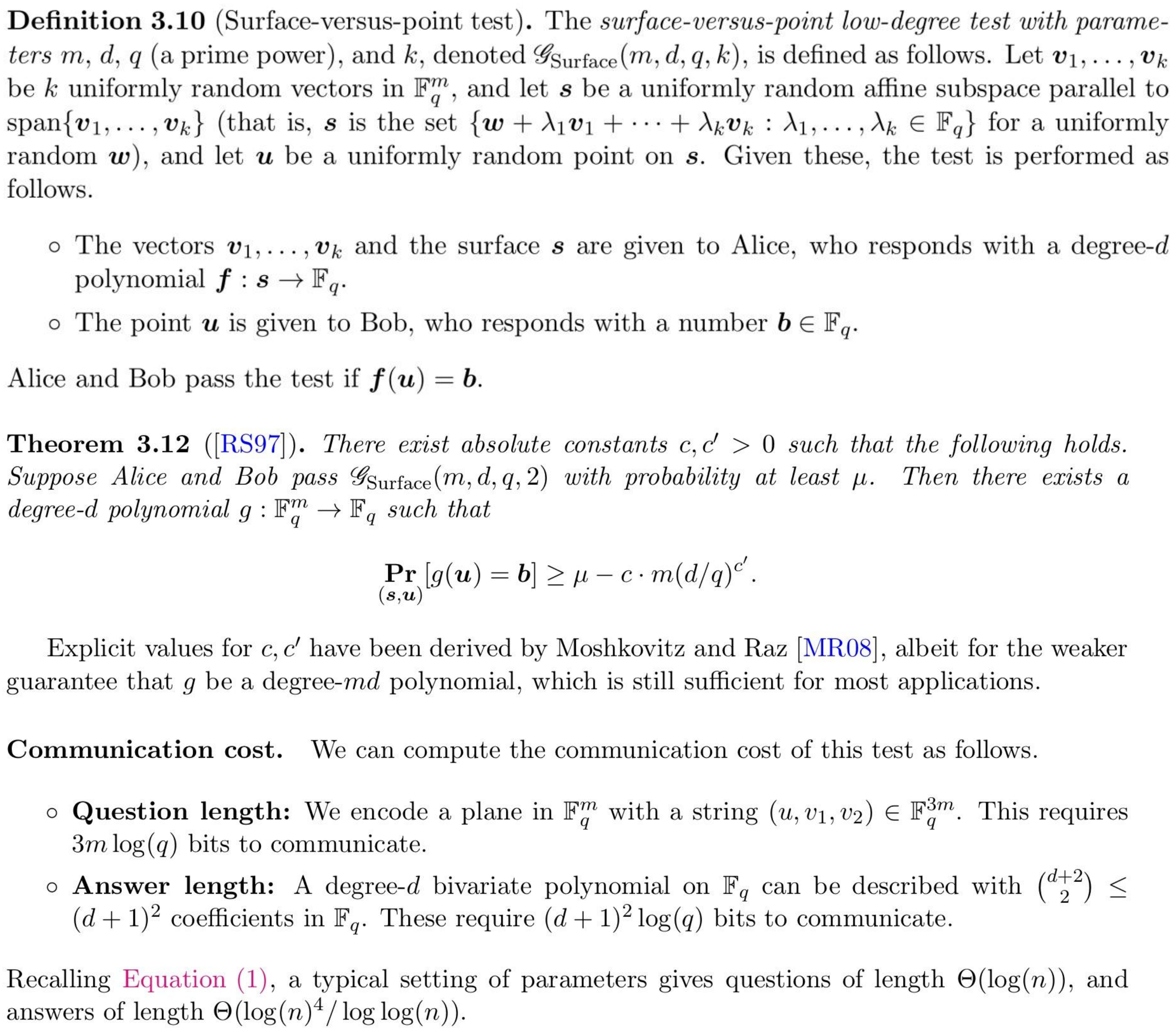
if we can show S-S-3SAT solved by MIP protocol with quantum strategy, it shows that NEEXP in MIP*.

That is, we will construct MIP* game that completes S-S-3SAT problem in **polynomial answer and question – time and length** with

completeness and soundness

1. SvP and Classical PCP Theorem

Low-degree Testing (SvP)



Definition 3.10 (Surface-versus-point test)**.** The *surface-versus-point low-degree test with parameters* m, d, q (a prime power), and k, denoted $\mathscr{G}_{\text{Surface}}(m, d, q, k)$, is defined as follows. Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k$ be k uniformly random vectors in $\mathbb{F}_q^m$, and let $\boldsymbol{s}$ be a uniformly random affine subspace parallel to $\text{span}\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$ (that is, $\boldsymbol{s}$ is the set $\{\boldsymbol{w} + \lambda_1 \boldsymbol{v}_1 + \cdots + \lambda_k \boldsymbol{v}_k : \lambda_1, \ldots, \lambda_k \in \mathbb{F}_q\}$ for a uniformly random $\boldsymbol{w}$), and let $\boldsymbol{u}$ be a uniformly random point on $\boldsymbol{s}$. Given these, the test is performed as follows.

- The vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k$ and the surface $\boldsymbol{s}$ are given to Alice, who responds with a degree-d polynomial $\boldsymbol{f} : \boldsymbol{s} \to \mathbb{F}_q$.
- The point $\boldsymbol{u}$ is given to Bob, who responds with a number $\boldsymbol{b} \in \mathbb{F}_q$.

Alice and Bob pass the test if $\boldsymbol{f}(\boldsymbol{u}) = \boldsymbol{b}$.

Theorem 3.12 ([RS97])**.** *There exist absolute constants* $c, c' > 0$ *such that the following holds. Suppose Alice and Bob pass* $\mathscr{G}_{\text{Surface}}(m, d, q, 2)$ *with probability at least* μ*. Then there exists a degree-d polynomial* $g : \mathbb{F}_q^m \to \mathbb{F}_q$ *such that*

$$\Pr_{(\boldsymbol{s},\boldsymbol{u})}[g(\boldsymbol{u}) = \boldsymbol{b}] \geq \mu - c \cdot m(d/q)^{c'}.$$

Explicit values for c, c' have been derived by Moshkovitz and Raz [MR08], albeit for the weaker guarantee that g be a degree-md polynomial, which is still sufficient for most applications.

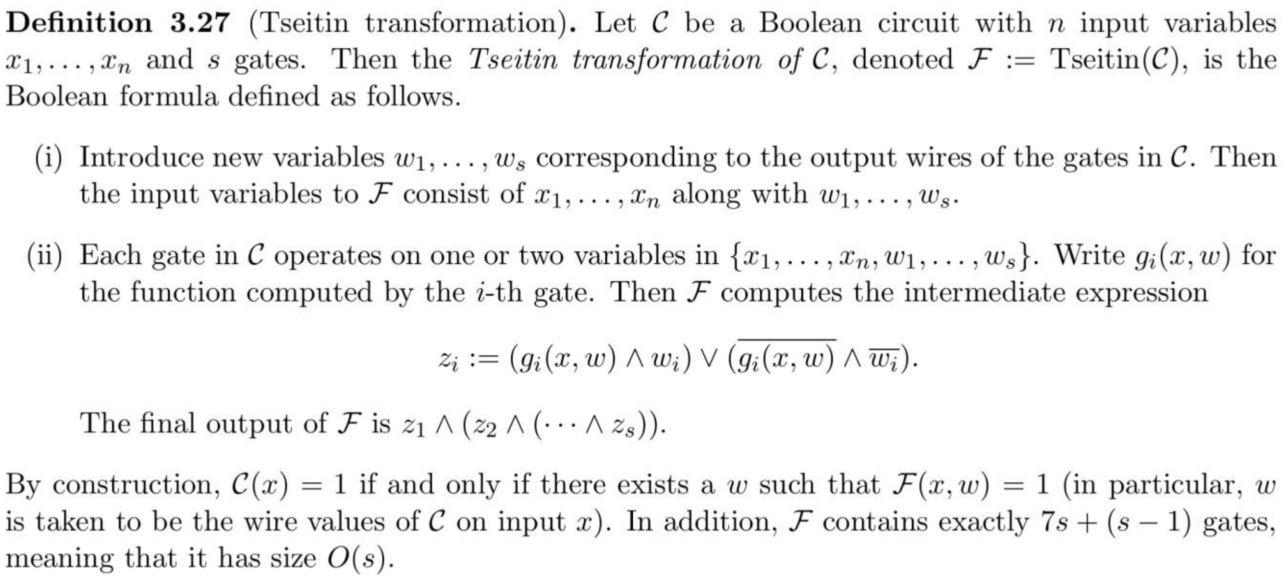
Communication cost. We can compute the communication cost of this test as follows.

- **Question length:** We encode a plane in $\mathbb{F}_q^m$ with a string $(u, v_1, v_2) \in \mathbb{F}_q^{3m}$. This requires $3m \log(q)$ bits to communicate.
- **Answer length:** A degree-d bivariate polynomial on $\mathbb{F}_q$ can be described with $\binom{d+2}{2} \leq (d+1)^2$ coefficients in $\mathbb{F}_q$. These require $(d+1)^2 \log(q)$ bits to communicate.

Recalling Equation (1), a typical setting of parameters gives questions of length $\Theta(\log(n))$, and answers of length $\Theta(\log(n)^4 / \log\log(n))$.

Existence of
global low-degree polynomial
of prover's process

Tseitin Transformation and Arithmetization



Definition 3.27 (Tseitin transformation)**.** Let $\mathcal{C}$ be a Boolean circuit with n input variables $x_1, \ldots, x_n$ and s gates. Then the *Tseitin transformation of* $\mathcal{C}$, denoted $\mathcal{F} := \text{Tseitin}(\mathcal{C})$, is the Boolean formula defined as follows.

(i) Introduce new variables $w_1, \ldots, w_s$ corresponding to the output wires of the gates in $\mathcal{C}$. Then the input variables to $\mathcal{F}$ consist of $x_1, \ldots, x_n$ along with $w_1, \ldots, w_s$.

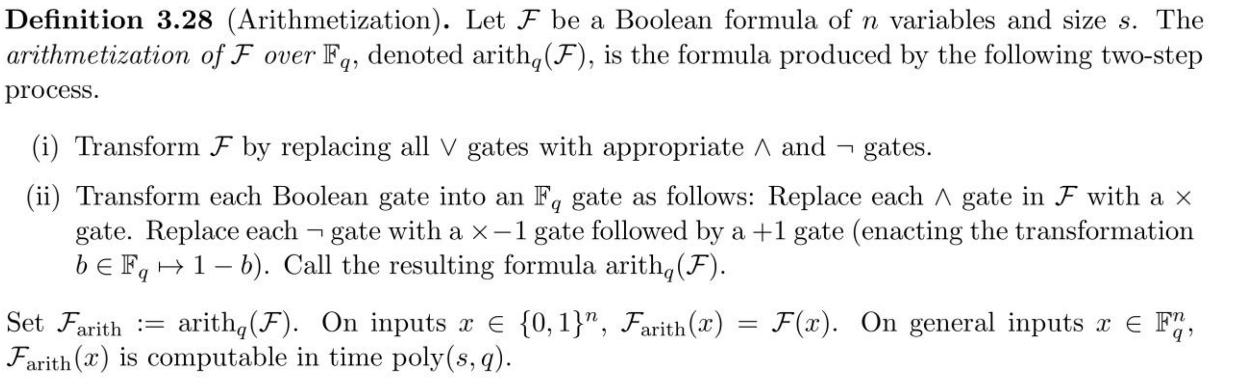
(ii) Each gate in $\mathcal{C}$ operates on one or two variables in $\{x_1, \ldots, x_n, w_1, \ldots, w_s\}$. Write $g_i(x, w)$ for the function computed by the i-th gate. Then $\mathcal{F}$ computes the intermediate expression

$$z_i := (g_i(x, w) \wedge w_i) \vee (\overline{g_i(x, w)} \wedge \overline{w_i}).$$

The final output of $\mathcal{F}$ is $z_1 \wedge (z_2 \wedge (\cdots \wedge z_s))$.

By construction, $\mathcal{C}(x) = 1$ if and only if there exists a w such that $\mathcal{F}(x, w) = 1$ (in particular, w is taken to be the wire values of $\mathcal{C}$ on input x). In addition, $\mathcal{F}$ contains exactly $7s + (s - 1)$ gates, meaning that it has size $O(s)$.

Boolean Circuit
➔ Boolean Formula
➔ Polynomial over {0,1}^k

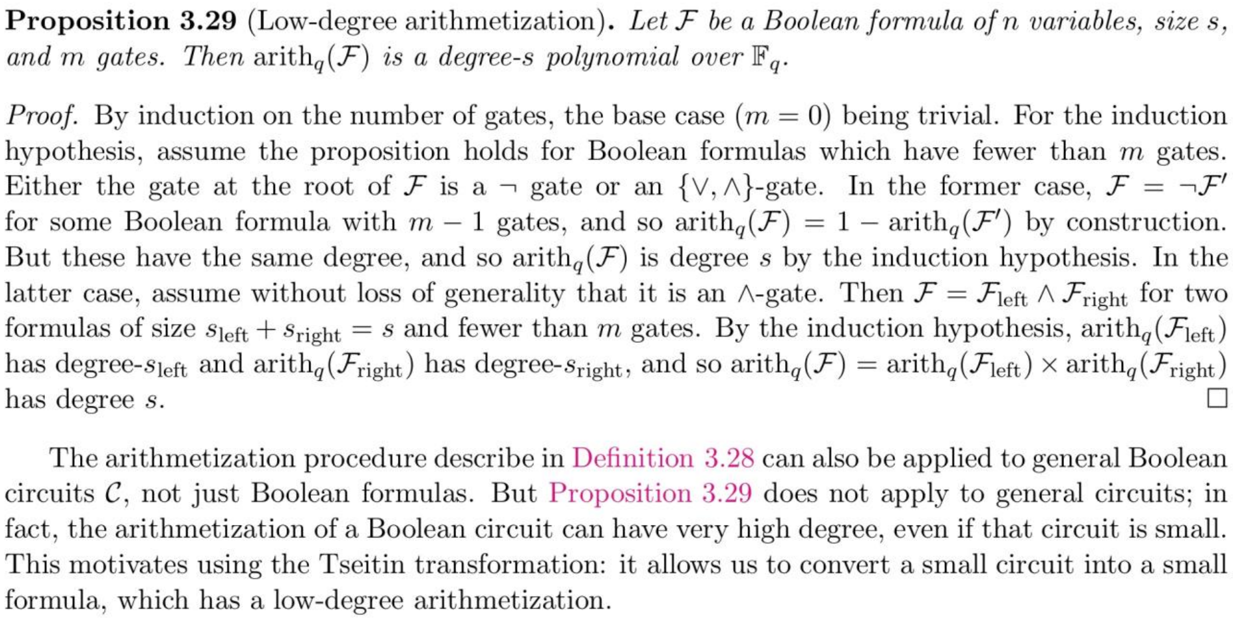


Definition 3.28 (Arithmetization)**.** Let $\mathcal{F}$ be a Boolean formula of n variables and size s. The *arithmetization of* $\mathcal{F}$ *over* $\mathbb{F}_q$, denoted $\text{arith}_q(\mathcal{F})$, is the formula produced by the following two-step process.

(i) Transform $\mathcal{F}$ by replacing all $\vee$ gates with appropriate $\wedge$ and $\neg$ gates.

(ii) Transform each Boolean gate into an $\mathbb{F}_q$ gate as follows: Replace each $\wedge$ gate in $\mathcal{F}$ with a $\times$ gate. Replace each $\neg$ gate with a $\times -1$ gate followed by a $+1$ gate (enacting the transformation $b \in \mathbb{F}_q \mapsto 1 - b$). Call the resulting formula $\text{arith}_q(\mathcal{F})$.

Set $\mathcal{F}_{\text{arith}} := \text{arith}_q(\mathcal{F})$. On inputs $x \in \{0,1\}^n$, $\mathcal{F}_{\text{arith}}(x) = \mathcal{F}(x)$. On general inputs $x \in \mathbb{F}_q^n$, $\mathcal{F}_{\text{arith}}(x)$ is computable in time $\text{poly}(s, q)$.



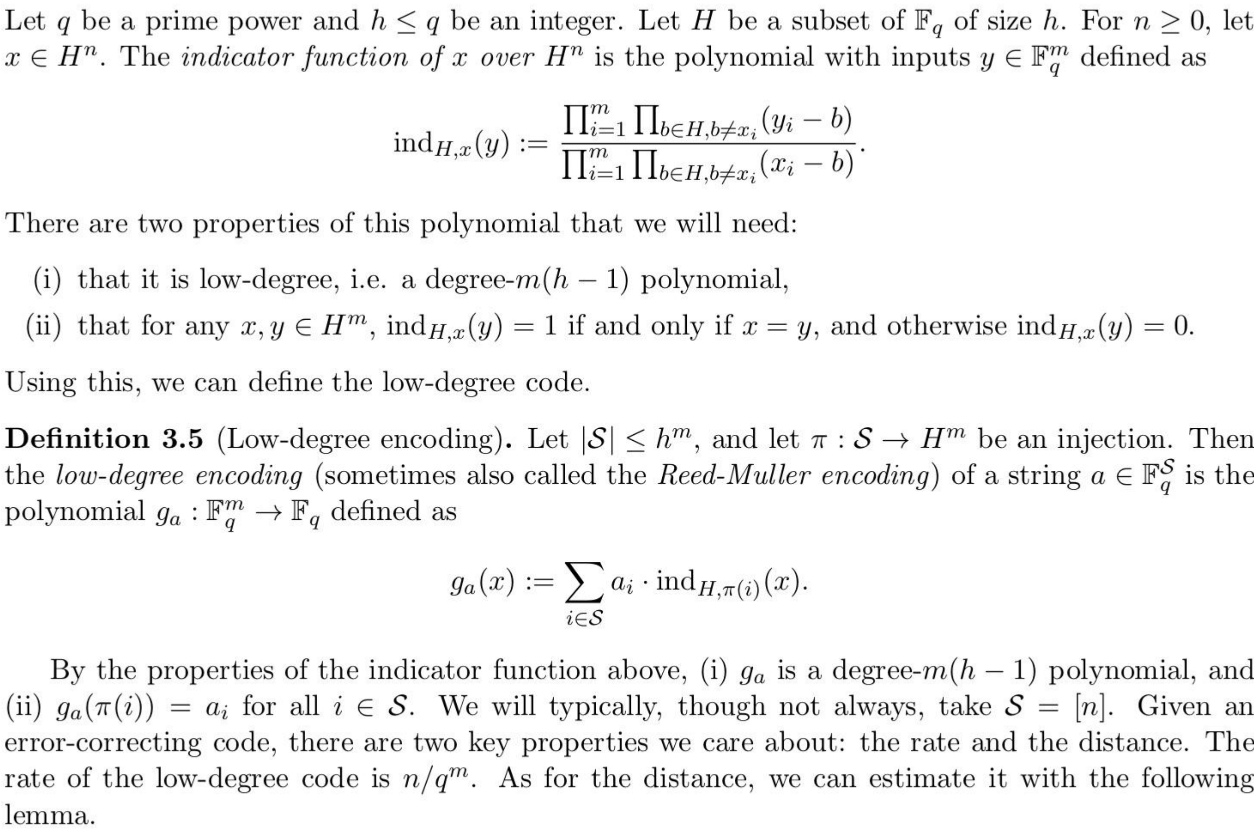
Proposition 3.29 (Low-degree arithmetization)**.** *Let* $\mathcal{F}$ *be a Boolean formula of* n *variables, size* s*, and* m *gates. Then* $\text{arith}_q(\mathcal{F})$ *is a degree-*s *polynomial over* $\mathbb{F}_q$.

Proof. By induction on the number of gates, the base case $(m = 0)$ being trivial. For the induction hypothesis, assume the proposition holds for Boolean formulas which have fewer than m gates. Either the gate at the root of $\mathcal{F}$ is a $\neg$ gate or an $\{\vee, \wedge\}$-gate. In the former case, $\mathcal{F} = \neg\mathcal{F}'$ for some Boolean formula with $m - 1$ gates, and so $\text{arith}_q(\mathcal{F}) = 1 - \text{arith}_q(\mathcal{F}')$ by construction. But these have the same degree, and so $\text{arith}_q(\mathcal{F})$ is degree s by the induction hypothesis. In the latter case, assume without loss of generality that it is an $\wedge$-gate. Then $\mathcal{F} = \mathcal{F}_{\text{left}} \wedge \mathcal{F}_{\text{right}}$ for two formulas of size $s_{\text{left}} + s_{\text{right}} = s$ and fewer than m gates. By the induction hypothesis, $\text{arith}_q(\mathcal{F}_{\text{left}})$ has degree-s_{left} and $\text{arith}_q(\mathcal{F}_{\text{right}})$ has degree-s_{right}, and so $\text{arith}_q(\mathcal{F}) = \text{arith}_q(\mathcal{F}_{\text{left}}) \times \text{arith}_q(\mathcal{F}_{\text{right}})$ has degree s. □

The arithmetization procedure describe in Definition 3.28 can also be applied to general Boolean circuits $\mathcal{C}$, not just Boolean formulas. But Proposition 3.29 does not apply to general circuits; in fact, the arithmetization of a Boolean circuit can have very high degree, even if that circuit is small. This motivates using the Tseitin transformation: it allows us to convert a small circuit into a small formula, which has a low-degree arithmetization.

Low Degree Encoding

Motivation:
Interpolation on finite field v.s.



Let q be a prime power and $h \le q$ be an integer. Let H be a subset of $\mathbb{F}_q$ of size h. For $n \ge 0$, let $x \in H^n$. The *indicator function of x over H^n* is the polynomial with inputs $y \in \mathbb{F}_q^m$ defined as

$$\mathrm{ind}_{H,x}(y) := \frac{\prod_{i=1}^{m} \prod_{b \in H, b \neq x_i} (y_i - b)}{\prod_{i=1}^{m} \prod_{b \in H, b \neq x_i} (x_i - b)}.$$

There are two properties of this polynomial that we will need:

(i) that it is low-degree, i.e. a degree-$m(h-1)$ polynomial,

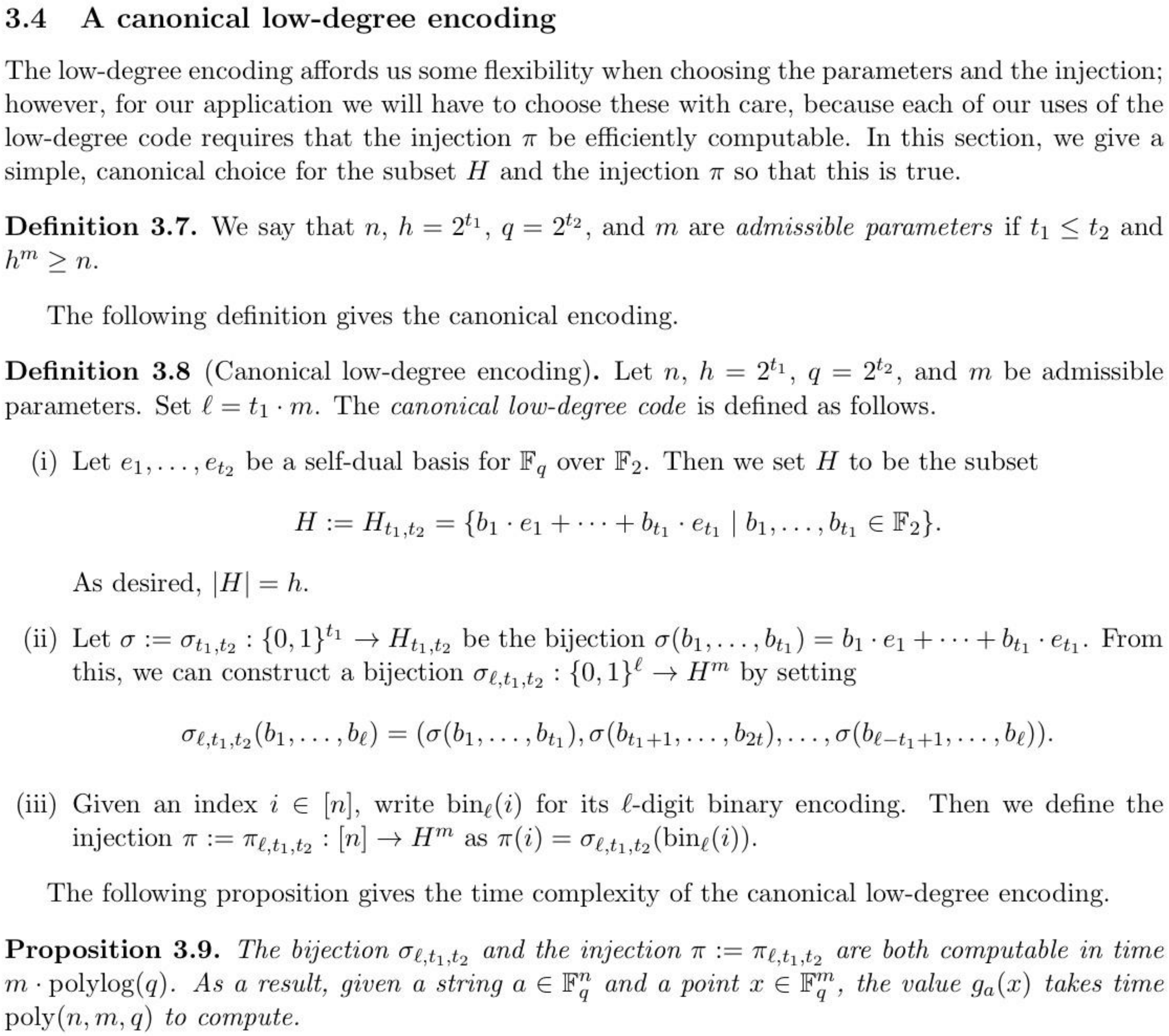
(ii) that for any $x, y \in H^m$, $\mathrm{ind}_{H,x}(y) = 1$ if and only if $x = y$, and otherwise $\mathrm{ind}_{H,x}(y) = 0$.

Using this, we can define the low-degree code.

Definition 3.5 (Low-degree encoding)**.** Let $|\mathcal{S}| \le h^m$, and let $\pi : \mathcal{S} \to H^m$ be an injection. Then the *low-degree encoding* (sometimes also called the *Reed-Muller encoding*) of a string $a \in \mathbb{F}_q^{\mathcal{S}}$ is the polynomial $g_a : \mathbb{F}_q^m \to \mathbb{F}_q$ defined as

$$g_a(x) := \sum_{i \in \mathcal{S}} a_i \cdot \mathrm{ind}_{H,\pi(i)}(x).$$

By the properties of the indicator function above, (i) g_a is a degree-$m(h-1)$ polynomial, and (ii) $g_a(\pi(i)) = a_i$ for all $i \in \mathcal{S}$. We will typically, though not always, take $\mathcal{S} = [n]$. Given an error-correcting code, there are two key properties we care about: the rate and the distance. The rate of the low-degree code is n/q^m. As for the distance, we can estimate it with the following lemma.



3.4 A canonical low-degree encoding

The low-degree encoding affords us some flexibility when choosing the parameters and the injection; however, for our application we will have to choose these with care, because each of our uses of the low-degree code requires that the injection π be efficiently computable. In this section, we give a simple, canonical choice for the subset H and the injection π so that this is true.

Definition 3.7. We say that n, $h = 2^{t_1}$, $q = 2^{t_2}$, and m are *admissible parameters* if $t_1 \le t_2$ and $h^m \ge n$.

The following definition gives the canonical encoding.

Definition 3.8 (Canonical low-degree encoding)**.** Let n, $h = 2^{t_1}$, $q = 2^{t_2}$, and m be admissible parameters. Set $\ell = t_1 \cdot m$. The *canonical low-degree code* is defined as follows.

(i) Let $e_1, \ldots, e_{t_2}$ be a self-dual basis for $\mathbb{F}_q$ over $\mathbb{F}_2$. Then we set H to be the subset

$$H := H_{t_1,t_2} = \{b_1 \cdot e_1 + \cdots + b_{t_1} \cdot e_{t_1} \mid b_1, \ldots, b_{t_1} \in \mathbb{F}_2\}.$$

As desired, $|H| = h$.

(ii) Let $\sigma := \sigma_{t_1,t_2} : \{0,1\}^{t_1} \to H_{t_1,t_2}$ be the bijection $\sigma(b_1, \ldots, b_{t_1}) = b_1 \cdot e_1 + \cdots + b_{t_1} \cdot e_{t_1}$. From this, we can construct a bijection $\sigma_{\ell,t_1,t_2} : \{0,1\}^{\ell} \to H^m$ by setting

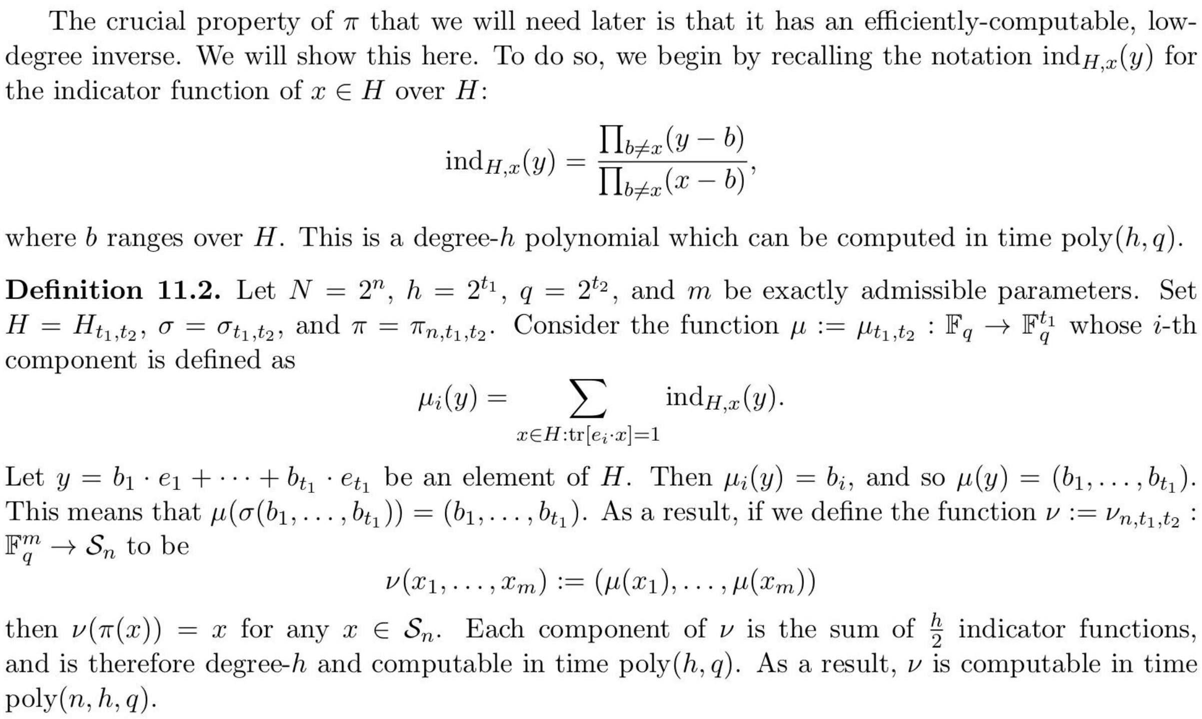
$$\sigma_{\ell,t_1,t_2}(b_1, \ldots, b_\ell) = (\sigma(b_1, \ldots, b_{t_1}), \sigma(b_{t_1+1}, \ldots, b_{2t}), \ldots, \sigma(b_{\ell-t_1+1}, \ldots, b_\ell)).$$

(iii) Given an index $i \in [n]$, write $\mathrm{bin}_\ell(i)$ for its ℓ-digit binary encoding. Then we define the injection $\pi := \pi_{\ell,t_1,t_2} : [n] \to H^m$ as $\pi(i) = \sigma_{\ell,t_1,t_2}(\mathrm{bin}_\ell(i))$.

The following proposition gives the time complexity of the canonical low-degree encoding.

Proposition 3.9. *The bijection σ_{ℓ,t_1,t_2} and the injection $\pi := \pi_{\ell,t_1,t_2}$ are both computable in time $m \cdot \mathrm{polylog}(q)$. As a result, given a string $a \in \mathbb{F}_q^n$ and a point $x \in \mathbb{F}_q^m$, the value $g_a(x)$ takes time $\mathrm{poly}(n, m, q)$ to compute.*

Idea of classical PCP



The crucial property of π that we will need later is that it has an efficiently-computable, low-degree inverse. We will show this here. To do so, we begin by recalling the notation $\mathrm{ind}_{H,x}(y)$ for the indicator function of $x \in H$ over H:

$$\mathrm{ind}_{H,x}(y) = \frac{\prod_{b \neq x}(y-b)}{\prod_{b \neq x}(x-b)},$$

where b ranges over H. This is a degree-h polynomial which can be computed in time $\mathrm{poly}(h,q)$.

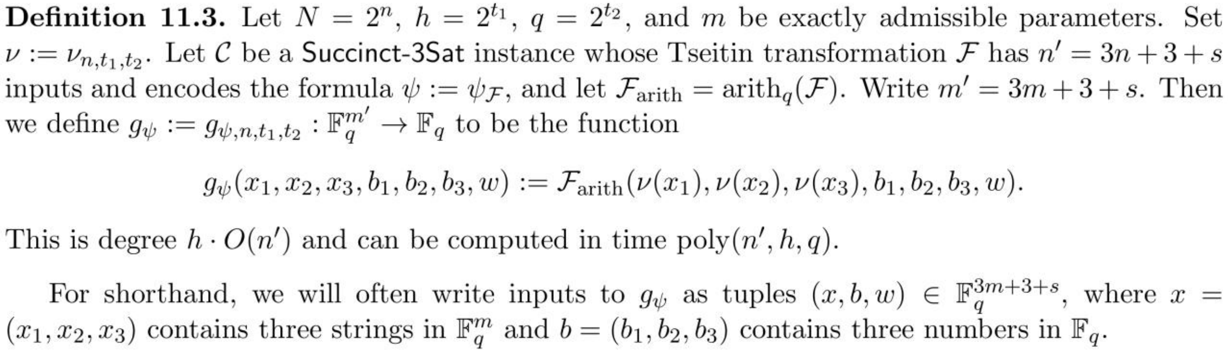
Definition 11.2. Let $N = 2^n$, $h = 2^{t_1}$, $q = 2^{t_2}$, and m be exactly admissible parameters. Set $H = H_{t_1,t_2}$, $\sigma = \sigma_{t_1,t_2}$, and $\pi = \pi_{n,t_1,t_2}$. Consider the function $\mu := \mu_{t_1,t_2} : \mathbb{F}_q \to \mathbb{F}_q^{t_1}$ whose i-th component is defined as

$$\mu_i(y) = \sum_{x \in H : \mathrm{tr}[e_i \cdot x] = 1} \mathrm{ind}_{H,x}(y).$$

Let $y = b_1 \cdot e_1 + \cdots + b_{t_1} \cdot e_{t_1}$ be an element of H. Then $\mu_i(y) = b_i$, and so $\mu(y) = (b_1, \ldots, b_{t_1})$. This means that $\mu(\sigma(b_1, \ldots, b_{t_1})) = (b_1, \ldots, b_{t_1})$. As a result, if we define the function $\nu := \nu_{n,t_1,t_2} : \mathbb{F}_q^m \to \mathcal{S}_n$ to be

$$\nu(x_1, \ldots, x_m) := (\mu(x_1), \ldots, \mu(x_m))$$

then $\nu(\pi(x)) = x$ for any $x \in \mathcal{S}_n$. Each component of ν is the sum of $\frac{h}{2}$ indicator functions, and is therefore degree-h and computable in time $\mathrm{poly}(h,q)$. As a result, ν is computable in time $\mathrm{poly}(n,h,q)$.

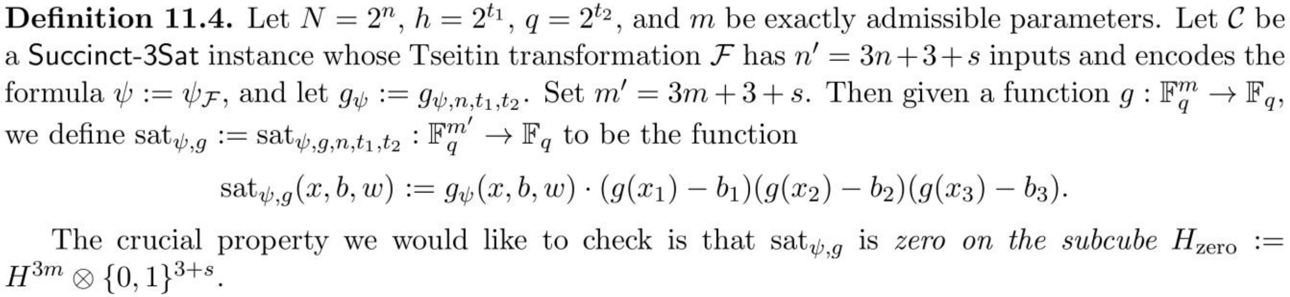


Definition 11.3. Let $N = 2^n$, $h = 2^{t_1}$, $q = 2^{t_2}$, and m be exactly admissible parameters. Set $\nu := \nu_{n,t_1,t_2}$. Let $\mathcal{C}$ be a Succinct-3Sat instance whose Tseitin transformation $\mathcal{F}$ has $n' = 3n+3+s$ inputs and encodes the formula $\psi := \psi_{\mathcal{F}}$, and let $\mathcal{F}_{\mathrm{arith}} = \mathrm{arith}_q(\mathcal{F})$. Write $m' = 3m+3+s$. Then we define $g_\psi := g_{\psi,n,t_1,t_2} : \mathbb{F}_q^{m'} \to \mathbb{F}_q$ to be the function

$$g_\psi(x_1, x_2, x_3, b_1, b_2, b_3, w) := \mathcal{F}_{\mathrm{arith}}(\nu(x_1), \nu(x_2), \nu(x_3), b_1, b_2, b_3, w).$$

This is degree $h \cdot O(n')$ and can be computed in time $\mathrm{poly}(n', h, q)$.

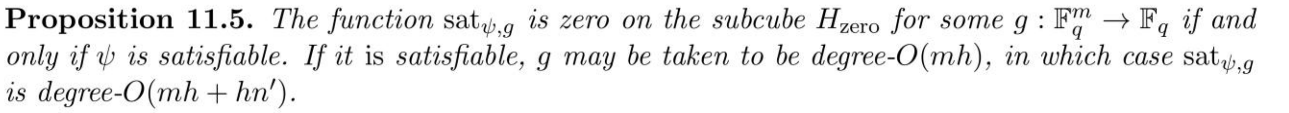
For shorthand, we will often write inputs to g_ψ as tuples $(x, b, w) \in \mathbb{F}_q^{3m+3+s}$, where $x = (x_1, x_2, x_3)$ contains three strings in $\mathbb{F}_q^m$ and $b = (b_1, b_2, b_3)$ contains three numbers in $\mathbb{F}_q$.



Definition 11.4. Let $N = 2^n$, $h = 2^{t_1}$, $q = 2^{t_2}$, and m be exactly admissible parameters. Let $\mathcal{C}$ be a Succinct-3Sat instance whose Tseitin transformation $\mathcal{F}$ has $n' = 3n+3+s$ inputs and encodes the formula $\psi := \psi_{\mathcal{F}}$, and let $g_\psi := g_{\psi,n,t_1,t_2}$. Set $m' = 3m+3+s$. Then given a function $g : \mathbb{F}_q^m \to \mathbb{F}_q$, we define $\mathrm{sat}_{\psi,g} := \mathrm{sat}_{\psi,g,n,t_1,t_2} : \mathbb{F}_q^{m'} \to \mathbb{F}_q$ to be the function

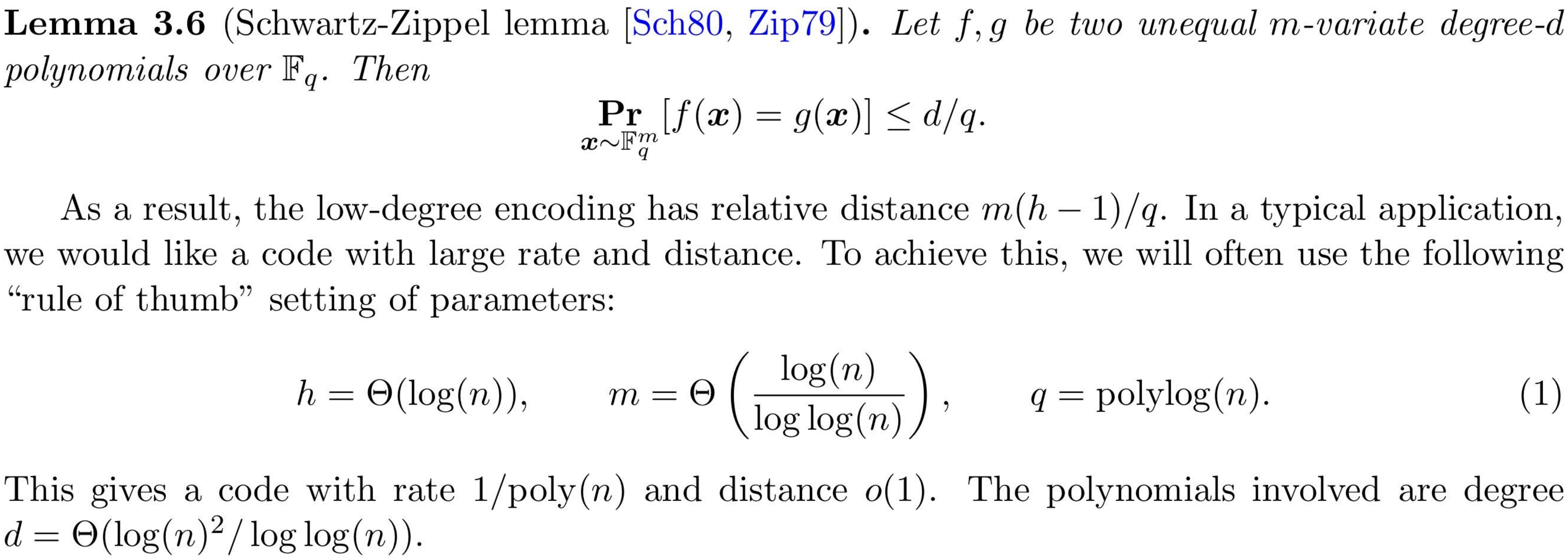
$$\mathrm{sat}_{\psi,g}(x, b, w) := g_\psi(x, b, w) \cdot (g(x_1) - b_1)(g(x_2) - b_2)(g(x_3) - b_3).$$

The crucial property we would like to check is that $\mathrm{sat}_{\psi,g}$ is *zero on the subcube* $H_{\mathrm{zero}} := H^{3m} \otimes \{0,1\}^{3+s}$.



Proposition 11.5. *The function $\mathrm{sat}_{\psi,g}$ is zero on the subcube H_{zero} for some $g : \mathbb{F}_q^m \to \mathbb{F}_q$ if and only if ψ is satisfiable. If it is satisfiable, g may be taken to be degree-$O(mh)$, in which case $\mathrm{sat}_{\psi,g}$ is degree-$O(mh + hn')$.*

Schwartz – Zippel Lemma



Lemma 3.6 (Schwartz-Zippel lemma [Sch80, Zip79])**.** *Let f, g be two unequal m-variate degree-d polynomials over $\mathbb{F}_q$. Then*

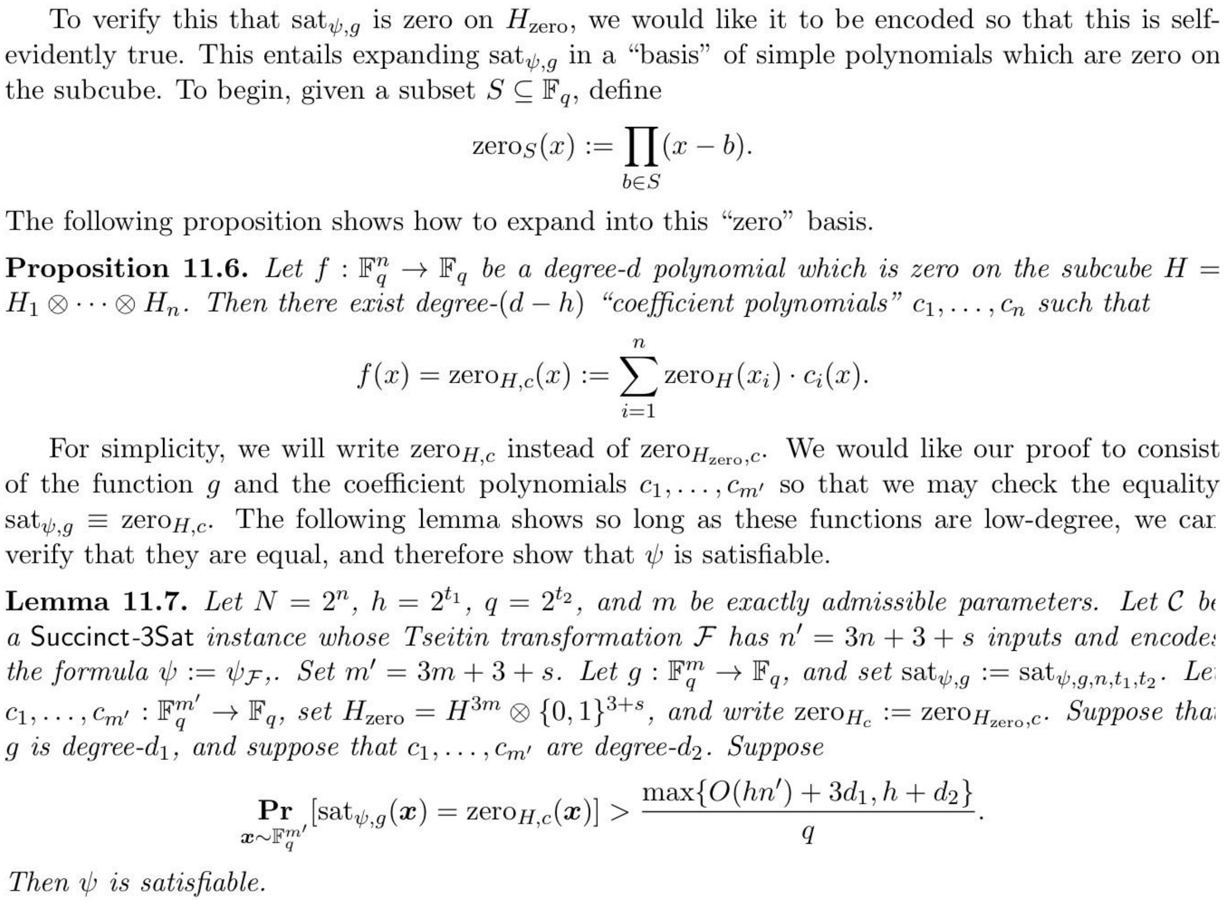
$$\mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathbb{F}_q^m} [f(\boldsymbol{x}) = g(\boldsymbol{x})] \leq d/q.$$

As a result, the low-degree encoding has relative distance $m(h-1)/q$. In a typical application, we would like a code with large rate and distance. To achieve this, we will often use the following "rule of thumb" setting of parameters:

$$h = \Theta(\log(n)), \qquad m = \Theta\left(\frac{\log(n)}{\log\log(n)}\right), \qquad q = \mathrm{polylog}(n). \tag{1}$$

This gives a code with rate $1/\mathrm{poly}(n)$ and distance $o(1)$. The polynomials involved are degree $d = \Theta(\log(n)^2/\log\log(n))$.

Idea of classical PCP



To verify this that $\mathrm{sat}_{\psi,g}$ is zero on H_{zero}, we would like it to be encoded so that this is self-evidently true. This entails expanding $\mathrm{sat}_{\psi,g}$ in a "basis" of simple polynomials which are zero on the subcube. To begin, given a subset $S \subseteq \mathbb{F}_q$, define

$$\mathrm{zero}_S(x) := \prod_{b \in S}(x - b).$$

The following proposition shows how to expand into this "zero" basis.

Proposition 11.6. *Let $f : \mathbb{F}_q^n \to \mathbb{F}_q$ be a degree-d polynomial which is zero on the subcube $H = H_1 \otimes \cdots \otimes H_n$. Then there exist degree-$(d-h)$ "coefficient polynomials" $c_1, \ldots, c_n$ such that*

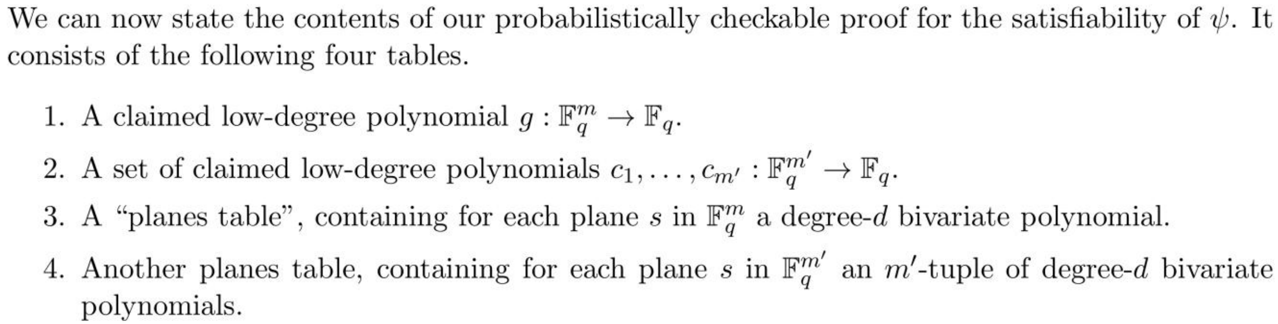
$$f(x) = \mathrm{zero}_{H,c}(x) := \sum_{i=1}^{n} \mathrm{zero}_H(x_i) \cdot c_i(x).$$

For simplicity, we will write $\mathrm{zero}_{H,c}$ instead of $\mathrm{zero}_{H_{\mathrm{zero}},c}$. We would like our proof to consist of the function g and the coefficient polynomials $c_1, \ldots, c_{m'}$ so that we may check the equality $\mathrm{sat}_{\psi,g} \equiv \mathrm{zero}_{H,c}$. The following lemma shows so long as these functions are low-degree, we car verify that they are equal, and therefore show that ψ is satisfiable.

Lemma 11.7. *Let $N = 2^n$, $h = 2^{t_1}$, $q = 2^{t_2}$, and m be exactly admissible parameters. Let $\mathcal{C}$ be a* Succinct-3Sat *instance whose Tseitin transformation $\mathcal{F}$ has $n' = 3n + 3 + s$ inputs and encode: the formula $\psi := \psi_{\mathcal{F}}$,. Set $m' = 3m + 3 + s$. Let $g : \mathbb{F}_q^m \to \mathbb{F}_q$, and set $\mathrm{sat}_{\psi,g} := \mathrm{sat}_{\psi,g,n,t_1,t_2}$. Le $c_1, \ldots, c_{m'} : \mathbb{F}_q^{m'} \to \mathbb{F}_q$, set $H_{\mathrm{zero}} = H^{3m} \otimes \{0,1\}^{3+s}$, and write $\mathrm{zero}_{H_c} := \mathrm{zero}_{H_{\mathrm{zero}},c}$. Suppose tha g is degree-d_1, and suppose that $c_1, \ldots, c_{m'}$ are degree-d_2. Suppose*

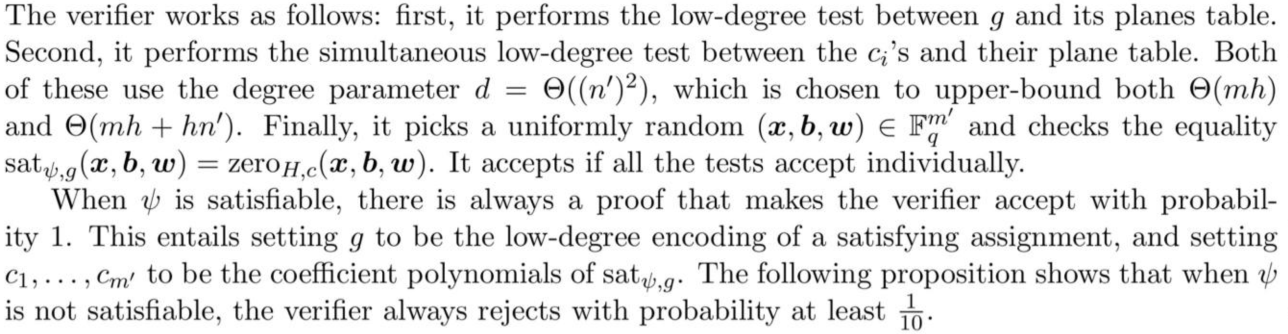
$$\mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathbb{F}_q^{m'}}[\mathrm{sat}_{\psi,g}(\boldsymbol{x}) = \mathrm{zero}_{H,c}(\boldsymbol{x})] > \frac{\max\{O(hn') + 3d_1, h + d_2\}}{q}.$$

Then ψ is satisfiable.



We can now state the contents of our probabilistically checkable proof for the satisfiability of ψ. It consists of the following four tables.

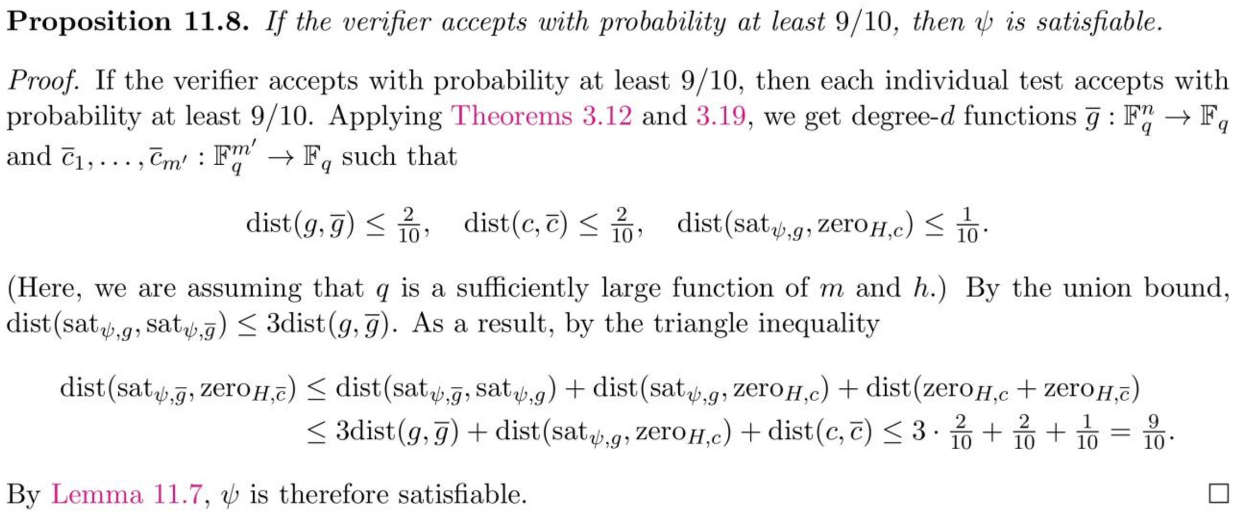
1. A claimed low-degree polynomial $g : \mathbb{F}_q^m \to \mathbb{F}_q$.
2. A set of claimed low-degree polynomials $c_1, \ldots, c_{m'} : \mathbb{F}_q^{m'} \to \mathbb{F}_q$.
3. A "planes table", containing for each plane s in $\mathbb{F}_q^m$ a degree-d bivariate polynomial.
4. Another planes table, containing for each plane s in $\mathbb{F}_q^{m'}$ an m'-tuple of degree-d bivariate polynomials.



The verifier works as follows: first, it performs the low-degree test between g and its planes table. Second, it performs the simultaneous low-degree test between the c_i's and their plane table. Both of these use the degree parameter $d = \Theta((n')^2)$, which is chosen to upper-bound both $\Theta(mh)$ and $\Theta(mh + hn')$. Finally, it picks a uniformly random $(\boldsymbol{x}, \boldsymbol{b}, \boldsymbol{w}) \in \mathbb{F}_q^{m'}$ and checks the equality $\mathrm{sat}_{\psi,g}(\boldsymbol{x}, \boldsymbol{b}, \boldsymbol{w}) = \mathrm{zero}_{H,c}(\boldsymbol{x}, \boldsymbol{b}, \boldsymbol{w})$. It accepts if all the tests accept individually.

When ψ is satisfiable, there is always a proof that makes the verifier accept with probability 1. This entails setting g to be the low-degree encoding of a satisfying assignment, and setting $c_1, \ldots, c_{m'}$ to be the coefficient polynomials of $\mathrm{sat}_{\psi,g}$. The following proposition shows that when ψ is not satisfiable, the verifier always rejects with probability at least $\frac{1}{10}$.

Idea of Classical PCP



Proposition 11.8. *If the verifier accepts with probability at least 9/10, then ψ is satisfiable.*

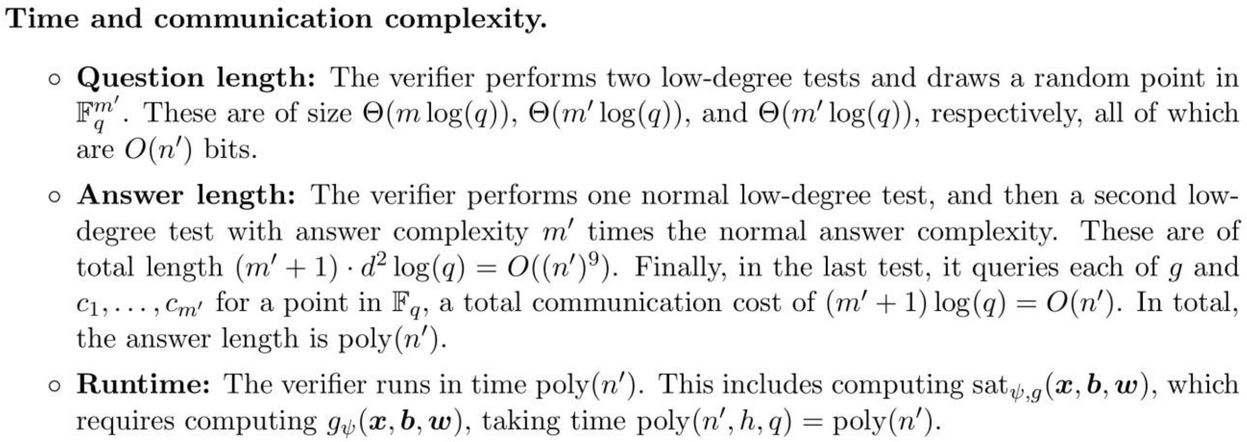
Proof. If the verifier accepts with probability at least 9/10, then each individual test accepts with probability at least 9/10. Applying Theorems 3.12 and 3.19, we get degree-d functions $\overline{g} : \mathbb{F}_q^n \to \mathbb{F}_q$ and $\overline{c}_1, \ldots, \overline{c}_{m'} : \mathbb{F}_q^{m'} \to \mathbb{F}_q$ such that

$$\mathrm{dist}(g, \overline{g}) \leq \tfrac{2}{10}, \quad \mathrm{dist}(c, \overline{c}) \leq \tfrac{2}{10}, \quad \mathrm{dist}(\mathrm{sat}_{\psi,g}, \mathrm{zero}_{H,c}) \leq \tfrac{1}{10}.$$

(Here, we are assuming that q is a sufficiently large function of m and h.) By the union bound, $\mathrm{dist}(\mathrm{sat}_{\psi,g}, \mathrm{sat}_{\psi,\overline{g}}) \leq 3\mathrm{dist}(g, \overline{g})$. As a result, by the triangle inequality

$$\begin{aligned}\mathrm{dist}(\mathrm{sat}_{\psi,\overline{g}}, \mathrm{zero}_{H,\overline{c}}) &\leq \mathrm{dist}(\mathrm{sat}_{\psi,\overline{g}}, \mathrm{sat}_{\psi,g}) + \mathrm{dist}(\mathrm{sat}_{\psi,g}, \mathrm{zero}_{H,c}) + \mathrm{dist}(\mathrm{zero}_{H,c} + \mathrm{zero}_{H,\overline{c}}) \\ &\leq 3\mathrm{dist}(g, \overline{g}) + \mathrm{dist}(\mathrm{sat}_{\psi,g}, \mathrm{zero}_{H,c}) + \mathrm{dist}(c, \overline{c}) \leq 3 \cdot \tfrac{2}{10} + \tfrac{2}{10} + \tfrac{1}{10} = \tfrac{9}{10}.\end{aligned}$$

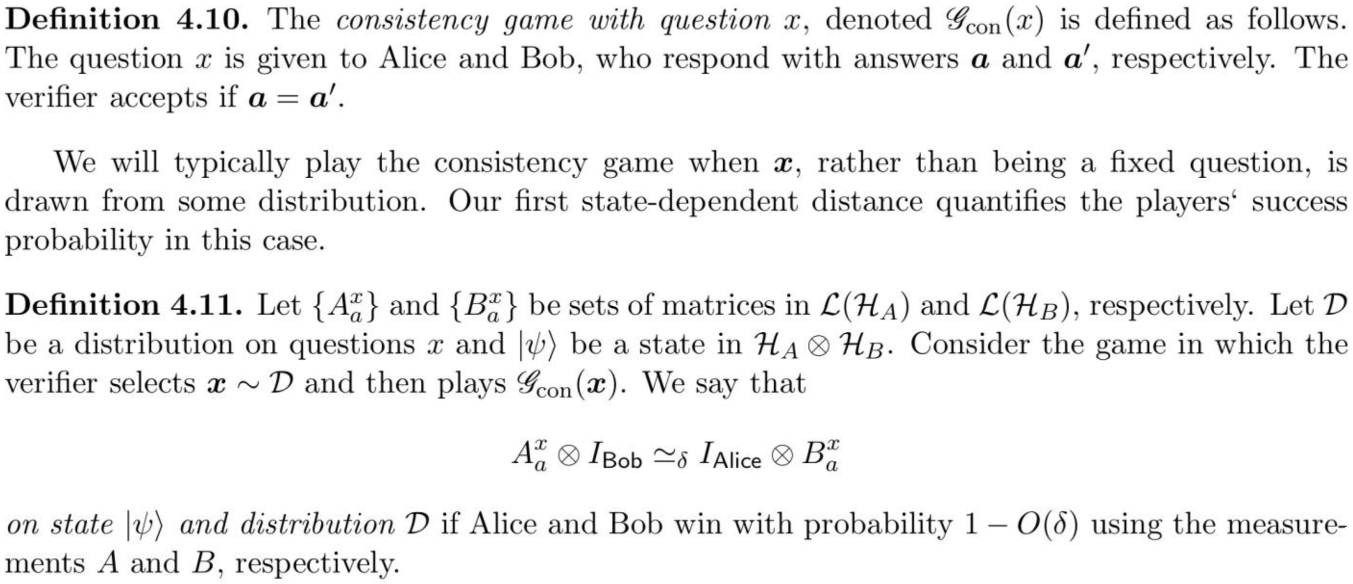
By Lemma 11.7, ψ is therefore satisfiable. □



Time and communication complexity.

- **Question length:** The verifier performs two low-degree tests and draws a random point in $\mathbb{F}_q^{m'}$. These are of size $\Theta(m \log(q))$, $\Theta(m' \log(q))$, and $\Theta(m' \log(q))$, respectively, all of which are $O(n')$ bits.
- **Answer length:** The verifier performs one normal low-degree test, and then a second low-degree test with answer complexity m' times the normal answer complexity. These are of total length $(m'+1) \cdot d^2 \log(q) = O((n')^9)$. Finally, in the last test, it queries each of g and $c_1, \ldots, c_{m'}$ for a point in $\mathbb{F}_q$, a total communication cost of $(m'+1)\log(q) = O(n')$. In total, the answer length is $\mathrm{poly}(n')$.
- **Runtime:** The verifier runs in time $\mathrm{poly}(n')$. This includes computing $\mathrm{sat}_{\psi,g}(\boldsymbol{x}, \boldsymbol{b}, \boldsymbol{w})$, which requires computing $g_\psi(\boldsymbol{x}, \boldsymbol{b}, \boldsymbol{w})$, taking time $\mathrm{poly}(n', h, q) = \mathrm{poly}(n')$.

State dependent distances



Definition 4.10. The *consistency game with question* x, denoted $\mathscr{G}_{\text{con}}(x)$ is defined as follows. The question x is given to Alice and Bob, who respond with answers $\boldsymbol{a}$ and $\boldsymbol{a}'$, respectively. The verifier accepts if $\boldsymbol{a} = \boldsymbol{a}'$.

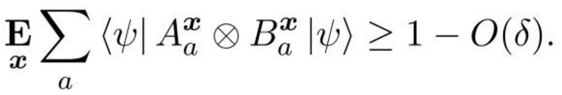
We will typically play the consistency game when $\boldsymbol{x}$, rather than being a fixed question, is drawn from some distribution. Our first state-dependent distance quantifies the players' success probability in this case.

Definition 4.11. Let $\{A_a^x\}$ and $\{B_a^x\}$ be sets of matrices in $\mathcal{L}(\mathcal{H}_A)$ and $\mathcal{L}(\mathcal{H}_B)$, respectively. Let $\mathcal{D}$ be a distribution on questions x and $|\psi\rangle$ be a state in $\mathcal{H}_A \otimes \mathcal{H}_B$. Consider the game in which the verifier selects $\boldsymbol{x} \sim \mathcal{D}$ and then plays $\mathscr{G}_{\text{con}}(\boldsymbol{x})$. We say that

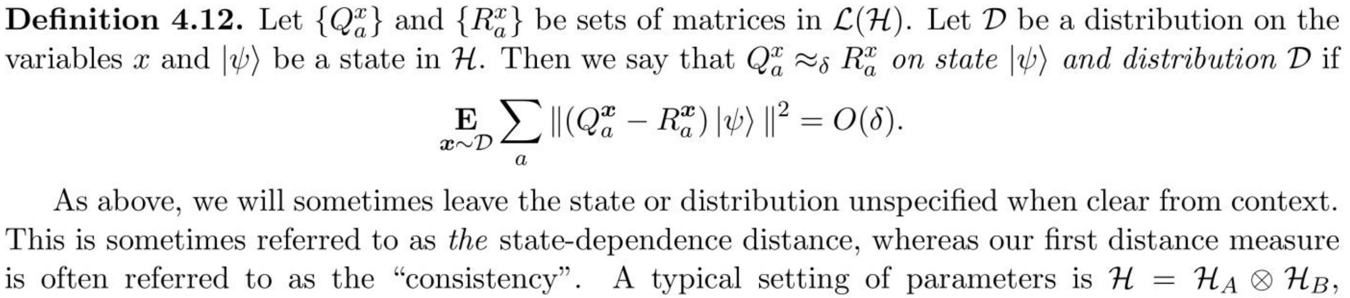
$$A_a^x \otimes I_{\mathsf{Bob}} \simeq_\delta I_{\mathsf{Alice}} \otimes B_a^x$$

on state $|\psi\rangle$ *and distribution* $\mathcal{D}$ if Alice and Bob win with probability $1 - O(\delta)$ using the measurements A and B, respectively.

Equivalent with:



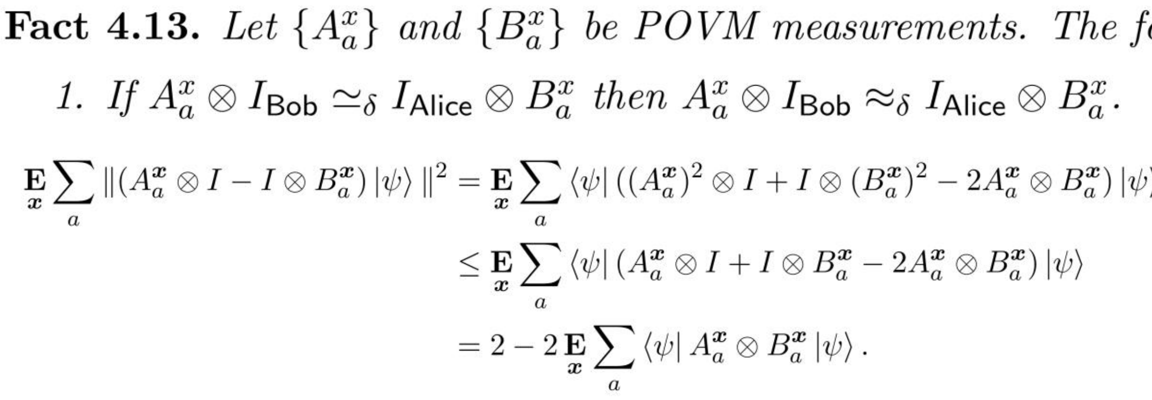
$$\mathop{\mathbf{E}}_{\boldsymbol{x}} \sum_a \langle\psi| A_a^{\boldsymbol{x}} \otimes B_a^{\boldsymbol{x}} |\psi\rangle \geq 1 - O(\delta).$$



Definition 4.12. Let $\{Q_a^x\}$ and $\{R_a^x\}$ be sets of matrices in $\mathcal{L}(\mathcal{H})$. Let $\mathcal{D}$ be a distribution on the variables x and $|\psi\rangle$ be a state in $\mathcal{H}$. Then we say that $Q_a^x \approx_\delta R_a^x$ *on state* $|\psi\rangle$ *and distribution* $\mathcal{D}$ if

$$\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{D}} \sum_a \| (Q_a^{\boldsymbol{x}} - R_a^{\boldsymbol{x}}) |\psi\rangle \|^2 = O(\delta).$$

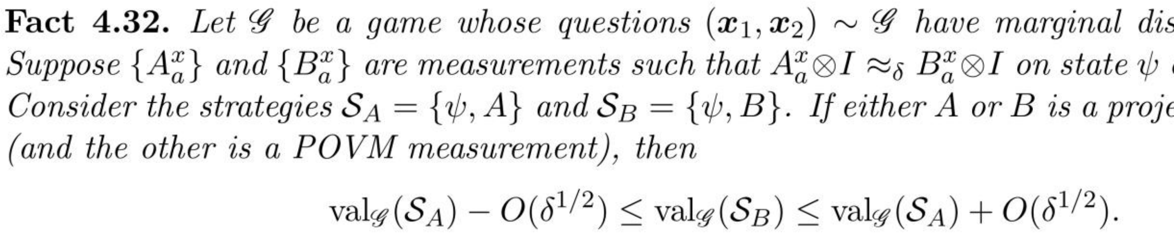
As above, we will sometimes leave the state or distribution unspecified when clear from context. This is sometimes referred to as *the* state-dependence distance, whereas our first distance measure is often referred to as the "consistency". A typical setting of parameters is $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$,



Fact 4.13. *Let* $\{A_a^x\}$ *and* $\{B_a^x\}$ *be POVM measurements. The f*

1. *If* $A_a^x \otimes I_{\mathsf{Bob}} \simeq_\delta I_{\mathsf{Alice}} \otimes B_a^x$ *then* $A_a^x \otimes I_{\mathsf{Bob}} \approx_\delta I_{\mathsf{Alice}} \otimes B_a^x$.

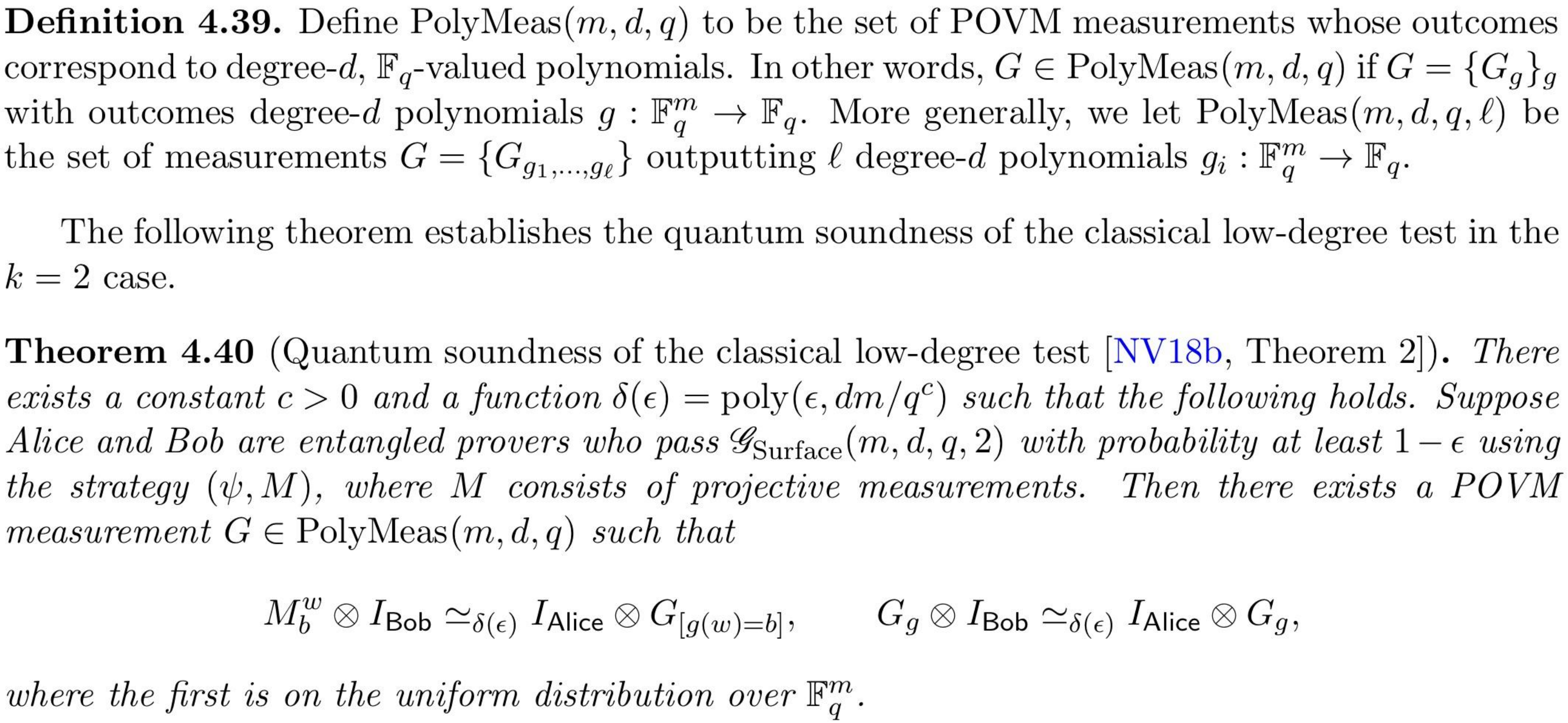
$$\begin{aligned}\mathop{\mathbf{E}}_{\boldsymbol{x}} \sum_a \| (A_a^{\boldsymbol{x}} \otimes I - I \otimes B_a^{\boldsymbol{x}}) |\psi\rangle \|^2 &= \mathop{\mathbf{E}}_{\boldsymbol{x}} \sum_a \langle\psi| ((A_a^{\boldsymbol{x}})^2 \otimes I + I \otimes (B_a^{\boldsymbol{x}})^2 - 2A_a^{\boldsymbol{x}} \otimes B_a^{\boldsymbol{x}}) |\psi \\ &\leq \mathop{\mathbf{E}}_{\boldsymbol{x}} \sum_a \langle\psi| (A_a^{\boldsymbol{x}} \otimes I + I \otimes B_a^{\boldsymbol{x}} - 2A_a^{\boldsymbol{x}} \otimes B_a^{\boldsymbol{x}}) |\psi\rangle \\ &= 2 - 2 \mathop{\mathbf{E}}_{\boldsymbol{x}} \sum_a \langle\psi| A_a^{\boldsymbol{x}} \otimes B_a^{\boldsymbol{x}} |\psi\rangle .\end{aligned}$$



Fact 4.32. *Let* $\mathscr{G}$ *be a game whose questions* $(\boldsymbol{x}_1, \boldsymbol{x}_2) \sim \mathscr{G}$ *have marginal dis*
Suppose $\{A_a^x\}$ *and* $\{B_a^x\}$ *are measurements such that* $A_a^x \otimes I \approx_\delta B_a^x \otimes I$ *on state* ψ
Consider the strategies $\mathcal{S}_A = \{\psi, A\}$ *and* $\mathcal{S}_B = \{\psi, B\}$. *If either* A *or* B *is a proje*
(and the other is a POVM measurement), then

$$\text{val}_{\mathscr{G}}(\mathcal{S}_A) - O(\delta^{1/2}) \leq \text{val}_{\mathscr{G}}(\mathcal{S}_B) \leq \text{val}_{\mathscr{G}}(\mathcal{S}_A) + O(\delta^{1/2}).$$

Quantum soundness of classical SvP



Definition 4.39. Define $\mathrm{PolyMeas}(m, d, q)$ to be the set of POVM measurements whose outcomes correspond to degree-d, $\mathbb{F}_q$-valued polynomials. In other words, $G \in \mathrm{PolyMeas}(m, d, q)$ if $G = \{G_g\}_g$ with outcomes degree-d polynomials $g : \mathbb{F}_q^m \to \mathbb{F}_q$. More generally, we let $\mathrm{PolyMeas}(m, d, q, \ell)$ be the set of measurements $G = \{G_{g_1,\ldots,g_\ell}\}$ outputting ℓ degree-d polynomials $g_i : \mathbb{F}_q^m \to \mathbb{F}_q$.

The following theorem establishes the quantum soundness of the classical low-degree test in the $k = 2$ case.

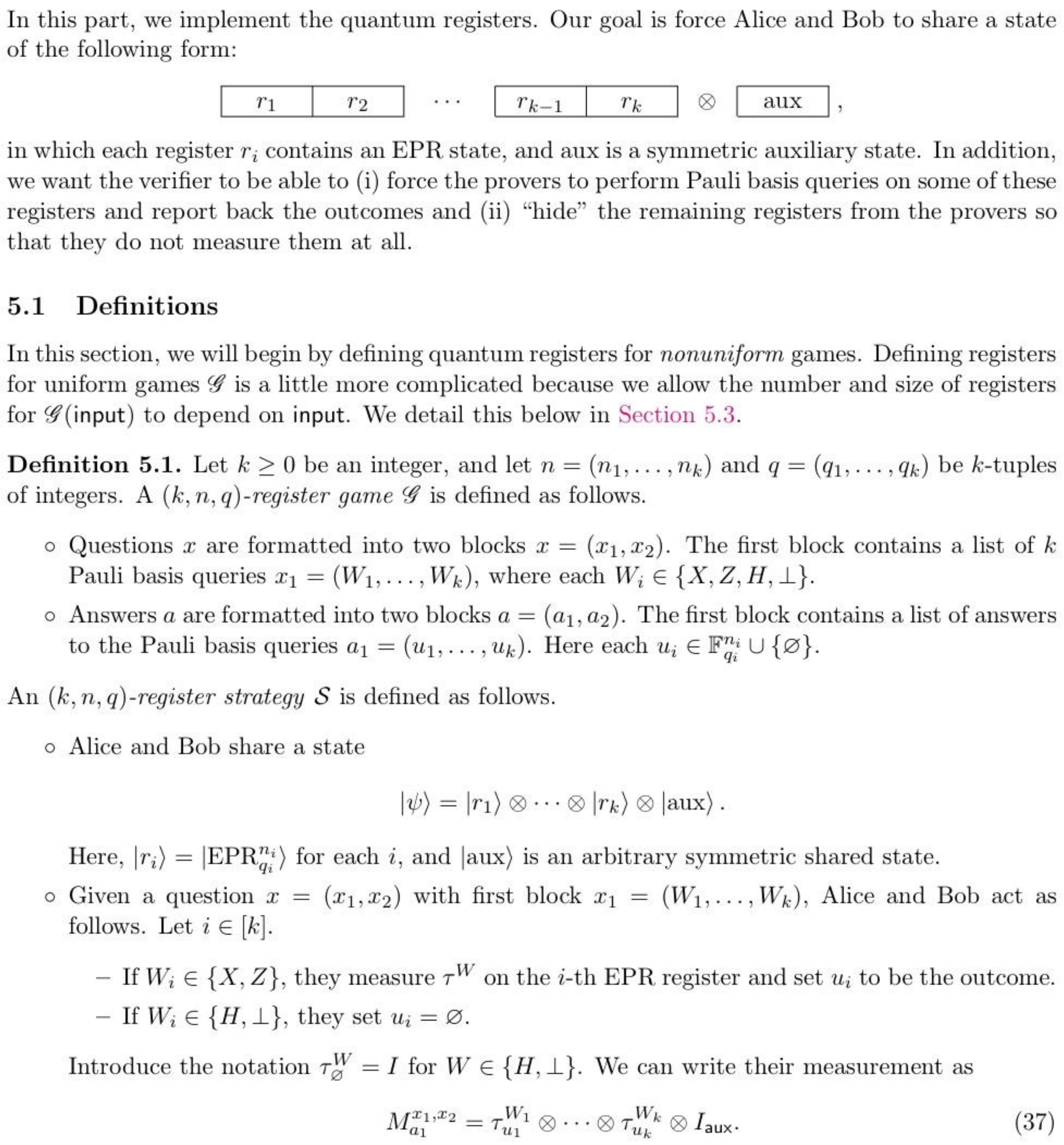
Theorem 4.40 (Quantum soundness of the classical low-degree test [NV18b, Theorem 2]). *There exists a constant $c > 0$ and a function $\delta(\epsilon) = \mathrm{poly}(\epsilon, dm/q^c)$ such that the following holds. Suppose Alice and Bob are entangled provers who pass $\mathscr{G}_{\mathrm{Surface}}(m, d, q, 2)$ with probability at least $1 - \epsilon$ using the strategy (ψ, M), where M consists of projective measurements. Then there exists a POVM measurement $G \in \mathrm{PolyMeas}(m, d, q)$ such that*

$$M_b^w \otimes I_{\mathsf{Bob}} \simeq_{\delta(\epsilon)} I_{\mathsf{Alice}} \otimes G_{[g(w)=b]}, \qquad G_g \otimes I_{\mathsf{Bob}} \simeq_{\delta(\epsilon)} I_{\mathsf{Alice}} \otimes G_g,$$

where the first is on the uniform distribution over $\mathbb{F}_q^m$.

2. Question Reduction

Register Game



In this part, we implement the quantum registers. Our goal is force Alice and Bob to share a state of the following form:

$$\boxed{r_1}\boxed{r_2} \quad \cdots \quad \boxed{r_{k-1}}\boxed{r_k} \otimes \boxed{\text{aux}} \, ,$$

in which each register r_i contains an EPR state, and aux is a symmetric auxiliary state. In addition, we want the verifier to be able to (i) force the provers to perform Pauli basis queries on some of these registers and report back the outcomes and (ii) "hide" the remaining registers from the provers so that they do not measure them at all.

5.1 Definitions

In this section, we will begin by defining quantum registers for *nonuniform* games. Defining registers for uniform games $\mathscr{G}$ is a little more complicated because we allow the number and size of registers for $\mathscr{G}(\mathsf{input})$ to depend on input. We detail this below in Section 5.3.

Definition 5.1. Let $k \geq 0$ be an integer, and let $n = (n_1, \dots, n_k)$ and $q = (q_1, \dots, q_k)$ be k-tuples of integers. A (k, n, q)*-register game* $\mathscr{G}$ is defined as follows.

- Questions x are formatted into two blocks $x = (x_1, x_2)$. The first block contains a list of k Pauli basis queries $x_1 = (W_1, \dots, W_k)$, where each $W_i \in \{X, Z, H, \perp\}$.
- Answers a are formatted into two blocks $a = (a_1, a_2)$. The first block contains a list of answers to the Pauli basis queries $a_1 = (u_1, \dots, u_k)$. Here each $u_i \in \mathbb{F}_{q_i}^{n_i} \cup \{\varnothing\}$.

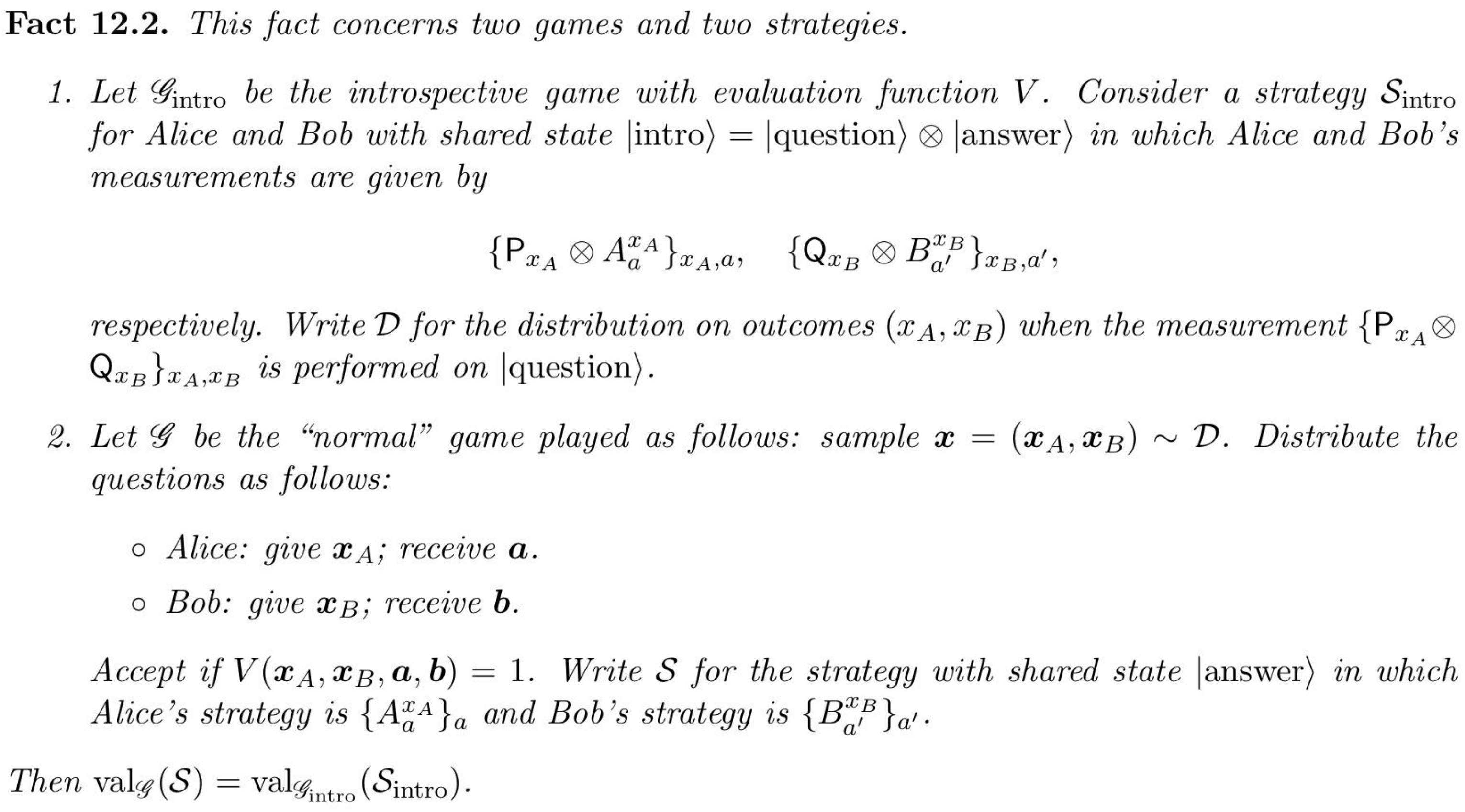
An (k, n, q)*-register strategy* $\mathcal{S}$ is defined as follows.

- Alice and Bob share a state
$$|\psi\rangle = |r_1\rangle \otimes \cdots \otimes |r_k\rangle \otimes |\text{aux}\rangle \, .$$
Here, $|r_i\rangle = |\text{EPR}_{q_i}^{n_i}\rangle$ for each i, and $|\text{aux}\rangle$ is an arbitrary symmetric shared state.
- Given a question $x = (x_1, x_2)$ with first block $x_1 = (W_1, \dots, W_k)$, Alice and Bob act as follows. Let $i \in [k]$.
  - If $W_i \in \{X, Z\}$, they measure τ^W on the i-th EPR register and set u_i to be the outcome.
  - If $W_i \in \{H, \perp\}$, they set $u_i = \varnothing$.

  Introduce the notation $\tau_\varnothing^W = I$ for $W \in \{H, \perp\}$. We can write their measurement as
$$M_{a_1}^{x_1,x_2} = \tau_{u_1}^{W_1} \otimes \cdots \otimes \tau_{u_k}^{W_k} \otimes I_{\mathsf{aux}}. \tag{37}$$

Introspection Games

Motivation : Since S-S-3SAT question is too big to satisfy interaction protocol.



Fact 12.2. *This fact concerns two games and two strategies.*

1. *Let* $\mathscr{G}_{\text{intro}}$ *be the introspective game with evaluation function* V. *Consider a strategy* $\mathcal{S}_{\text{intro}}$ *for Alice and Bob with shared state* $|\text{intro}\rangle = |\text{question}\rangle \otimes |\text{answer}\rangle$ *in which Alice and Bob's measurements are given by*

$$\{\mathsf{P}_{x_A} \otimes A_a^{x_A}\}_{x_A,a}, \quad \{\mathsf{Q}_{x_B} \otimes B_{a'}^{x_B}\}_{x_B,a'},$$

respectively. Write $\mathcal{D}$ *for the distribution on outcomes* (x_A, x_B) *when the measurement* $\{\mathsf{P}_{x_A} \otimes \mathsf{Q}_{x_B}\}_{x_A,x_B}$ *is performed on* $|\text{question}\rangle$.

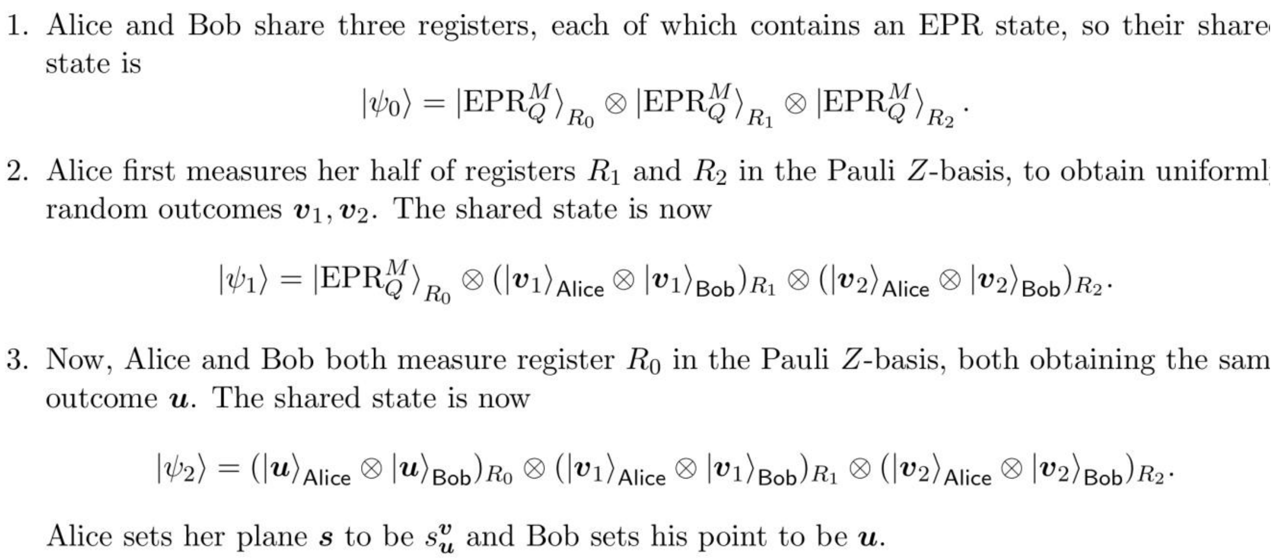
2. *Let* $\mathscr{G}$ *be the "normal" game played as follows: sample* $\boldsymbol{x} = (\boldsymbol{x}_A, \boldsymbol{x}_B) \sim \mathcal{D}$. *Distribute the questions as follows:*

   - *Alice: give* $\boldsymbol{x}_A$*; receive* $\boldsymbol{a}$.
   - *Bob: give* $\boldsymbol{x}_B$*; receive* $\boldsymbol{b}$.

   Accept if $V(\boldsymbol{x}_A, \boldsymbol{x}_B, \boldsymbol{a}, \boldsymbol{b}) = 1$. *Write* $\mathcal{S}$ *for the strategy with shared state* $|\text{answer}\rangle$ *in which Alice's strategy is* $\{A_a^{x_A}\}_a$ *and Bob's strategy is* $\{B_{a'}^{x_B}\}_{a'}$.

Then $\text{val}_{\mathscr{G}}(\mathcal{S}) = \text{val}_{\mathscr{G}_{\text{intro}}}(\mathcal{S}_{\text{intro}})$.

Introspective SvP



1. Alice and Bob share three registers, each of which contains an EPR state, so their share state is

$$|\psi_0\rangle = |\mathrm{EPR}_Q^M\rangle_{R_0} \otimes |\mathrm{EPR}_Q^M\rangle_{R_1} \otimes |\mathrm{EPR}_Q^M\rangle_{R_2}\,.$$

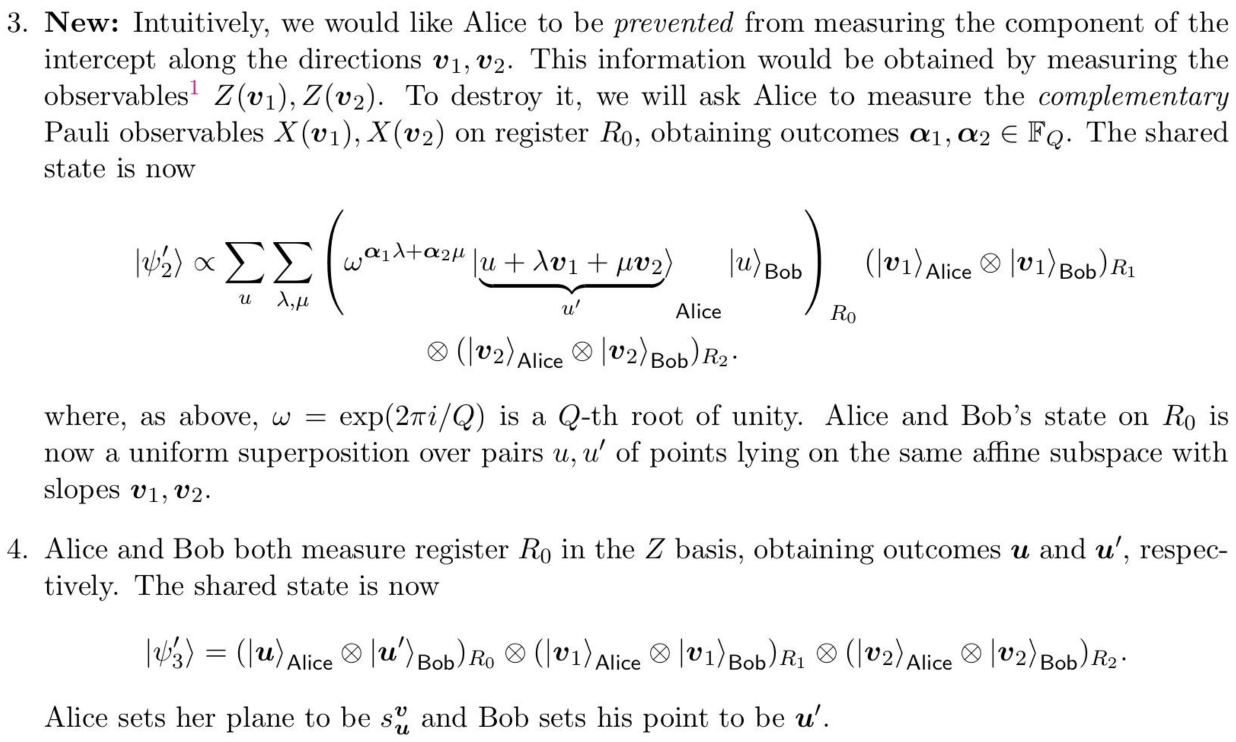
2. Alice first measures her half of registers R_1 and R_2 in the Pauli Z-basis, to obtain uniforml random outcomes $\boldsymbol{v}_1, \boldsymbol{v}_2$. The shared state is now

$$|\psi_1\rangle = |\mathrm{EPR}_Q^M\rangle_{R_0} \otimes (|\boldsymbol{v}_1\rangle_{\mathsf{Alice}} \otimes |\boldsymbol{v}_1\rangle_{\mathsf{Bob}})_{R_1} \otimes (|\boldsymbol{v}_2\rangle_{\mathsf{Alice}} \otimes |\boldsymbol{v}_2\rangle_{\mathsf{Bob}})_{R_2}.$$

3. Now, Alice and Bob both measure register R_0 in the Pauli Z-basis, both obtaining the sam outcome $\boldsymbol{u}$. The shared state is now

$$|\psi_2\rangle = (|\boldsymbol{u}\rangle_{\mathsf{Alice}} \otimes |\boldsymbol{u}\rangle_{\mathsf{Bob}})_{R_0} \otimes (|\boldsymbol{v}_1\rangle_{\mathsf{Alice}} \otimes |\boldsymbol{v}_1\rangle_{\mathsf{Bob}})_{R_1} \otimes (|\boldsymbol{v}_2\rangle_{\mathsf{Alice}} \otimes |\boldsymbol{v}_2\rangle_{\mathsf{Bob}})_{R_2}.$$

Alice sets her plane $\boldsymbol{s}$ to be $s_{\boldsymbol{u}}^{\boldsymbol{v}}$ and Bob sets his point to be $\boldsymbol{u}$.



3. **New:** Intuitively, we would like Alice to be *prevented* from measuring the component of the intercept along the directions $\boldsymbol{v}_1, \boldsymbol{v}_2$. This information would be obtained by measuring the observables[1] $Z(\boldsymbol{v}_1), Z(\boldsymbol{v}_2)$. To destroy it, we will ask Alice to measure the *complementary* Pauli observables $X(\boldsymbol{v}_1), X(\boldsymbol{v}_2)$ on register R_0, obtaining outcomes $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2 \in \mathbb{F}_Q$. The shared state is now

$$|\psi_2'\rangle \propto \sum_u \sum_{\lambda,\mu} \Big(\omega^{\boldsymbol{\alpha}_1\lambda + \boldsymbol{\alpha}_2\mu} \underbrace{|u + \lambda\boldsymbol{v}_1 + \mu\boldsymbol{v}_2\rangle}_{u'}{}_{\mathsf{Alice}} \quad |u\rangle_{\mathsf{Bob}} \Big)_{R_0} \quad (|\boldsymbol{v}_1\rangle_{\mathsf{Alice}} \otimes |\boldsymbol{v}_1\rangle_{\mathsf{Bob}})_{R_1}$$
$$\otimes (|\boldsymbol{v}_2\rangle_{\mathsf{Alice}} \otimes |\boldsymbol{v}_2\rangle_{\mathsf{Bob}})_{R_2}.$$

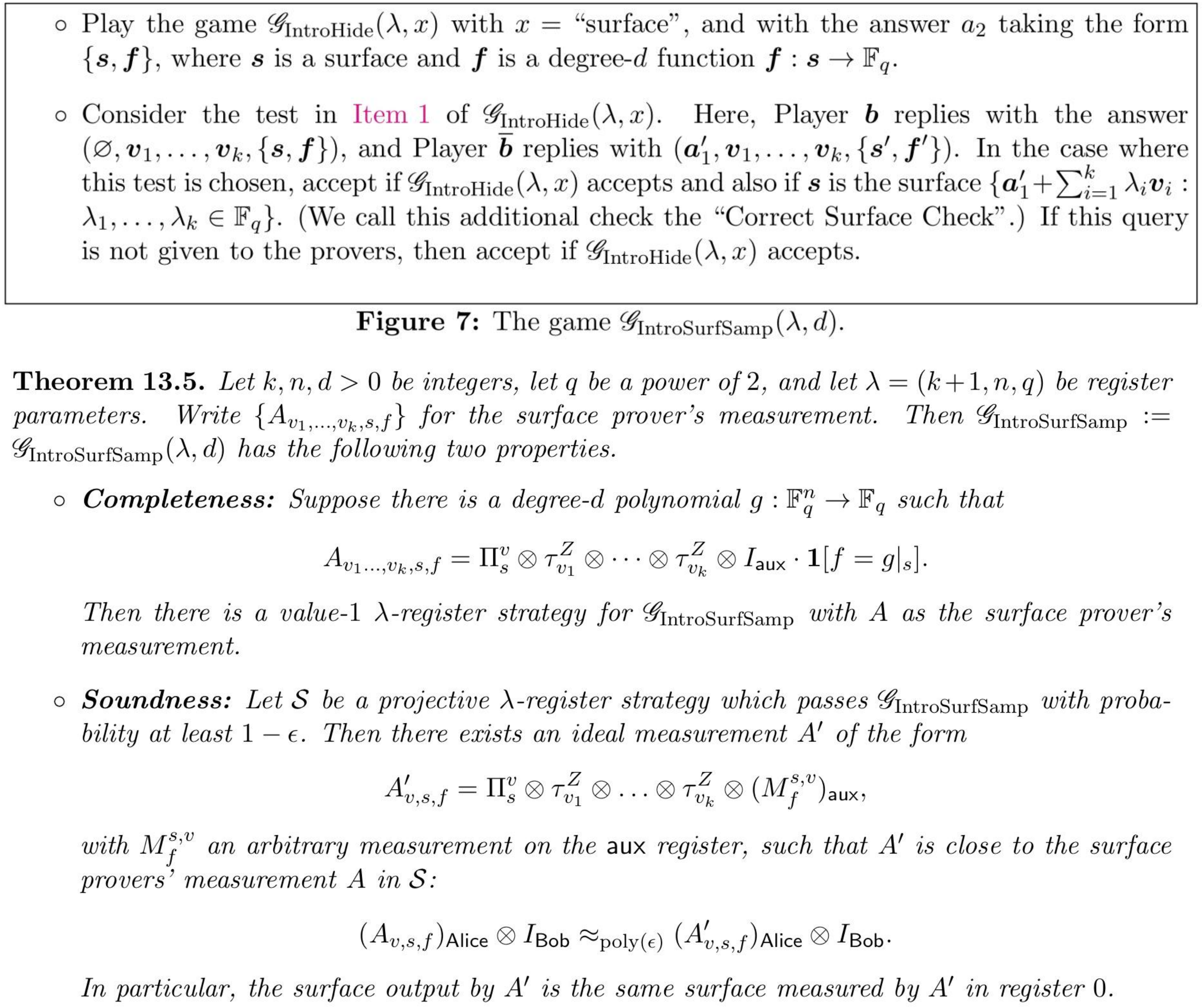
where, as above, $\omega = \exp(2\pi i/Q)$ is a Q-th root of unity. Alice and Bob's state on R_0 is now a uniform superposition over pairs u, u' of points lying on the same affine subspace with slopes $\boldsymbol{v}_1, \boldsymbol{v}_2$.

4. Alice and Bob both measure register R_0 in the Z basis, obtaining outcomes $\boldsymbol{u}$ and $\boldsymbol{u}'$, respectively. The shared state is now

$$|\psi_3'\rangle = (|\boldsymbol{u}\rangle_{\mathsf{Alice}} \otimes |\boldsymbol{u}'\rangle_{\mathsf{Bob}})_{R_0} \otimes (|\boldsymbol{v}_1\rangle_{\mathsf{Alice}} \otimes |\boldsymbol{v}_1\rangle_{\mathsf{Bob}})_{R_1} \otimes (|\boldsymbol{v}_2\rangle_{\mathsf{Alice}} \otimes |\boldsymbol{v}_2\rangle_{\mathsf{Bob}})_{R_2}.$$

Alice sets her plane to be $s_{\boldsymbol{u}}^{\boldsymbol{v}}$ and Bob sets his point to be $\boldsymbol{u}'$.

Introspective SvP



- Play the game $\mathscr{G}_{\text{IntroHide}}(\lambda, x)$ with $x =$ "surface", and with the answer a_2 taking the form $\{\boldsymbol{s}, \boldsymbol{f}\}$, where $\boldsymbol{s}$ is a surface and $\boldsymbol{f}$ is a degree-d function $\boldsymbol{f} : \boldsymbol{s} \to \mathbb{F}_q$.
- Consider the test in Item 1 of $\mathscr{G}_{\text{IntroHide}}(\lambda, x)$. Here, Player $\boldsymbol{b}$ replies with the answer $(\varnothing, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_k, \{\boldsymbol{s}, \boldsymbol{f}\})$, and Player $\overline{\boldsymbol{b}}$ replies with $(\boldsymbol{a}'_1, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_k, \{\boldsymbol{s}', \boldsymbol{f}'\})$. In the case where this test is chosen, accept if $\mathscr{G}_{\text{IntroHide}}(\lambda, x)$ accepts and also if $\boldsymbol{s}$ is the surface $\{\boldsymbol{a}'_1 + \sum_{i=1}^{k} \lambda_i \boldsymbol{v}_i : \lambda_1, \ldots, \lambda_k \in \mathbb{F}_q\}$. (We call this additional check the "Correct Surface Check".) If this query is not given to the provers, then accept if $\mathscr{G}_{\text{IntroHide}}(\lambda, x)$ accepts.

Figure 7: The game $\mathscr{G}_{\text{IntroSurfSamp}}(\lambda, d)$.

Theorem 13.5. *Let $k, n, d > 0$ be integers, let q be a power of 2, and let $\lambda = (k+1, n, q)$ be register parameters. Write $\{A_{v_1,\ldots,v_k,s,f}\}$ for the surface prover's measurement. Then $\mathscr{G}_{\text{IntroSurfSamp}} := \mathscr{G}_{\text{IntroSurfSamp}}(\lambda, d)$ has the following two properties.*

- ***Completeness:*** *Suppose there is a degree-d polynomial $g : \mathbb{F}_q^n \to \mathbb{F}_q$ such that*

$$A_{v_1\ldots,v_k,s,f} = \Pi_s^v \otimes \tau_{v_1}^Z \otimes \cdots \otimes \tau_{v_k}^Z \otimes I_{\mathsf{aux}} \cdot \mathbf{1}[f = g|_s].$$

  Then there is a value-1 λ-register strategy for $\mathscr{G}_{\text{IntroSurfSamp}}$ with A as the surface prover's measurement.

- ***Soundness:*** *Let $\mathcal{S}$ be a projective λ-register strategy which passes $\mathscr{G}_{\text{IntroSurfSamp}}$ with probability at least $1 - \epsilon$. Then there exists an ideal measurement A' of the form*

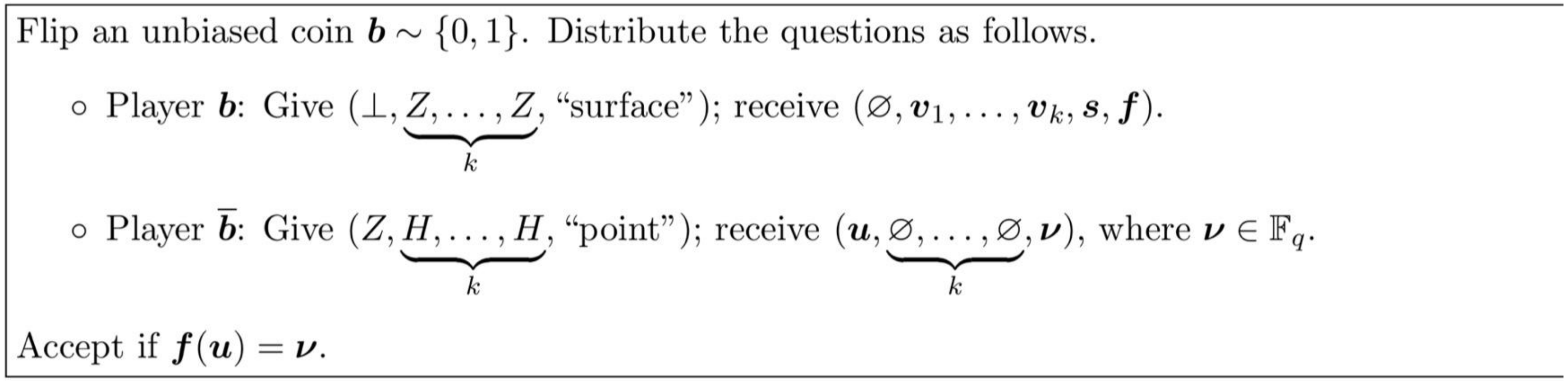
$$A'_{v,s,f} = \Pi_s^v \otimes \tau_{v_1}^Z \otimes \ldots \otimes \tau_{v_k}^Z \otimes (M_f^{s,v})_{\mathsf{aux}},$$

  with $M_f^{s,v}$ an arbitrary measurement on the aux *register, such that A' is close to the surface provers' measurement A in $\mathcal{S}$:*

$$(A_{v,s,f})_{\mathsf{Alice}} \otimes I_{\mathsf{Bob}} \approx_{\text{poly}(\epsilon)} (A'_{v,s,f})_{\mathsf{Alice}} \otimes I_{\mathsf{Bob}}.$$

  In particular, the surface output by A' is the same surface measured by A' in register 0.

Introspective SvP

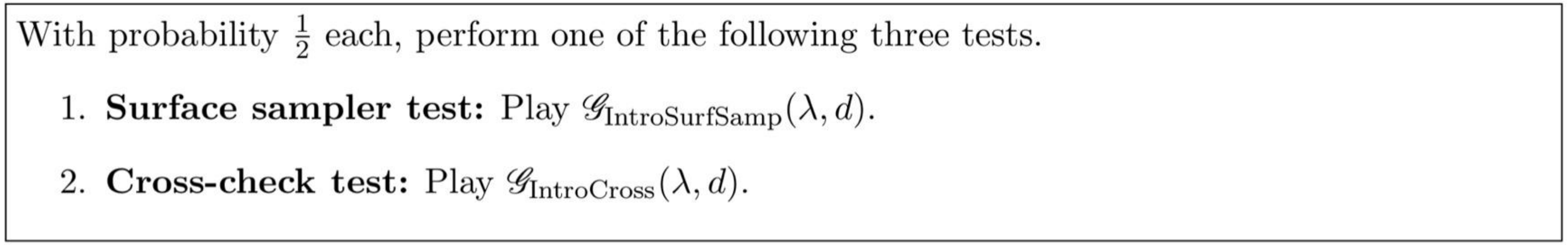


Flip an unbiased coin $\boldsymbol{b} \sim \{0,1\}$. Distribute the questions as follows.

- ○ Player $\boldsymbol{b}$: Give $(\perp, \underbrace{Z, \ldots, Z}_{k}, \text{“surface”})$; receive $(\varnothing, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_k, \boldsymbol{s}, \boldsymbol{f})$.
- ○ Player $\overline{\boldsymbol{b}}$: Give $(Z, \underbrace{H, \ldots, H}_{k}, \text{“point”})$; receive $(\boldsymbol{u}, \underbrace{\varnothing, \ldots, \varnothing}_{k}, \boldsymbol{\nu})$, where $\boldsymbol{\nu} \in \mathbb{F}_q$.

Accept if $\boldsymbol{f}(\boldsymbol{u}) = \boldsymbol{\nu}$.

Figure 8: The game $\mathscr{G}_{\text{IntroCross}}(\lambda, d)$.

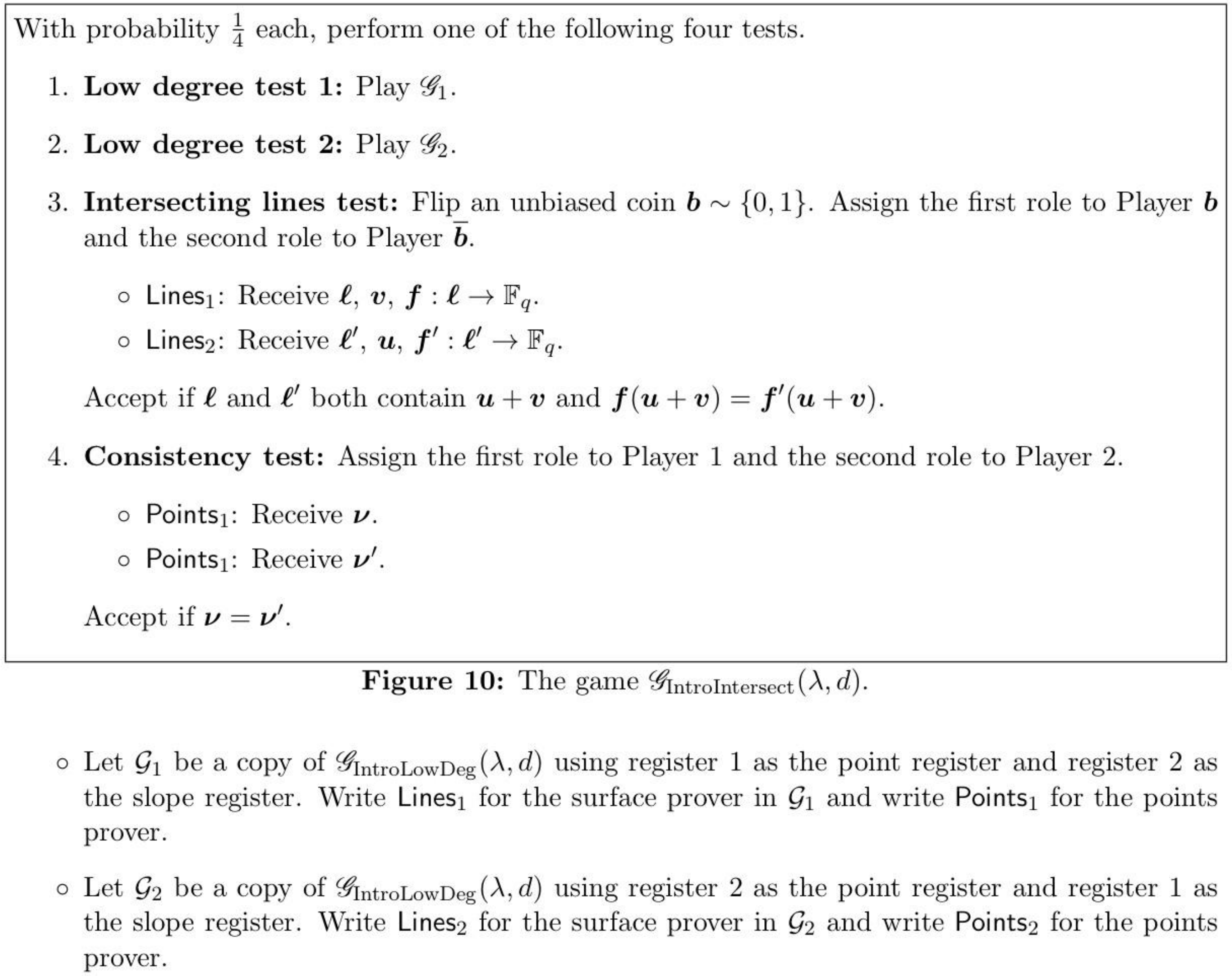


With probability $\frac{1}{2}$ each, perform one of the following three tests.

1. **Surface sampler test:** Play $\mathscr{G}_{\text{IntroSurfSamp}}(\lambda, d)$.
2. **Cross-check test:** Play $\mathscr{G}_{\text{IntroCross}}(\lambda, d)$.

Figure 9: The game $\mathscr{G}_{\text{IntroLowDeg}}(\lambda, d)$.

Introspective Intersecting Lines Test



With probability $\frac{1}{4}$ each, perform one of the following four tests.

1. **Low degree test 1:** Play $\mathscr{G}_1$.
2. **Low degree test 2:** Play $\mathscr{G}_2$.
3. **Intersecting lines test:** Flip an unbiased coin $\boldsymbol{b} \sim \{0,1\}$. Assign the first role to Player $\boldsymbol{b}$ and the second role to Player $\overline{\boldsymbol{b}}$.
   - Lines_1: Receive $\boldsymbol{\ell}$, $\boldsymbol{v}$, $\boldsymbol{f} : \boldsymbol{\ell} \to \mathbb{F}_q$.
   - Lines_2: Receive $\boldsymbol{\ell}'$, $\boldsymbol{u}$, $\boldsymbol{f}' : \boldsymbol{\ell}' \to \mathbb{F}_q$.

   Accept if $\boldsymbol{\ell}$ and $\boldsymbol{\ell}'$ both contain $\boldsymbol{u} + \boldsymbol{v}$ and $\boldsymbol{f}(\boldsymbol{u} + \boldsymbol{v}) = \boldsymbol{f}'(\boldsymbol{u} + \boldsymbol{v})$.
4. **Consistency test:** Assign the first role to Player 1 and the second role to Player 2.
   - Points_1: Receive $\boldsymbol{\nu}$.
   - Points_1: Receive $\boldsymbol{\nu}'$.

   Accept if $\boldsymbol{\nu} = \boldsymbol{\nu}'$.

Figure 10: The game $\mathscr{G}_{\text{IntroIntersect}}(\lambda, d)$.

- Let $\mathcal{G}_1$ be a copy of $\mathscr{G}_{\text{IntroLowDeg}}(\lambda, d)$ using register 1 as the point register and register 2 as the slope register. Write Lines_1 for the surface prover in $\mathcal{G}_1$ and write Points_1 for the points prover.
- Let $\mathcal{G}_2$ be a copy of $\mathscr{G}_{\text{IntroLowDeg}}(\lambda, d)$ using register 2 as the point register and register 1 as the slope register. Write Lines_2 for the surface prover in $\mathcal{G}_2$ and write Points_2 for the points prover.

Introspective Formula Game

$$|\psi\rangle = (|\mathrm{EPR}_q^m\rangle_1) \otimes |\mathrm{EPR}_q^m\rangle_2 \otimes |\mathrm{EPR}_q^m\rangle_3 \otimes |\mathrm{EPR}_q^{3+s}\rangle_4)_{\mathsf{SuperReg1}} \\ \otimes (|\mathrm{EPR}_q^{m'}\rangle_5)_{\mathsf{SuperReg2}} \otimes (|\mathrm{EPR}_q^{m'}\rangle_6)_{\mathsf{SuperReg3}} \otimes |\mathrm{aux}\rangle_{\mathsf{aux}}\,.$$

Definition 15.1. Let $\mathcal{C}_{\mathrm{inst}}$ be a size-s_{inst} instance of the Succinct-Succinct-3Sat problem.

1. Let $\mathcal{C}$ be the size-s Succinct-3Sat instance it succinctly represents. This circuit takes inputs i, j, k, each of some length n, and bits b_1, b_2, b_3. Then s and n can both be trivially upper-bounded by $N := 2^{s_{\mathrm{inst}}}$.

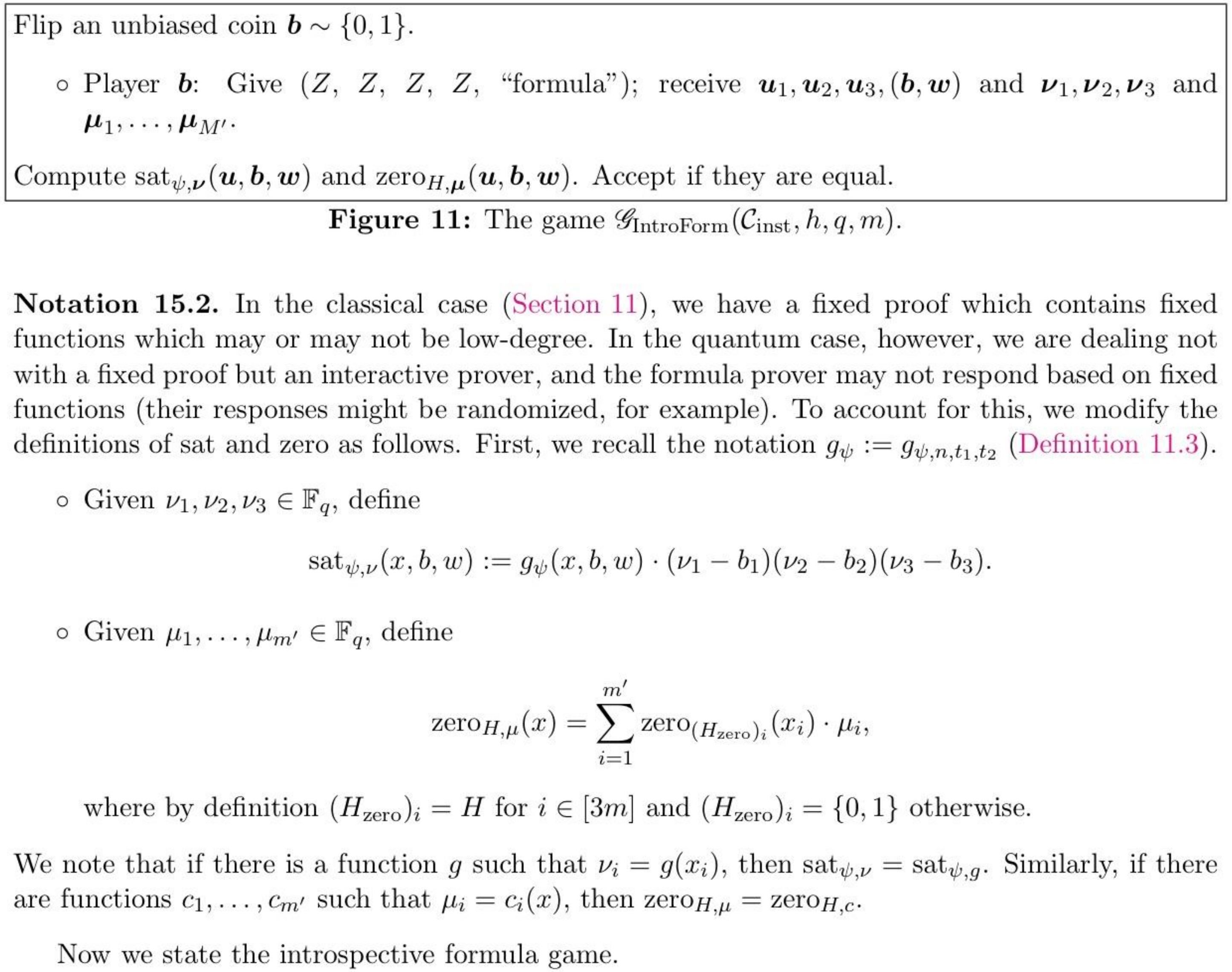
2. Consider a new circuit $\mathcal{C}_{\mathrm{pad}}$ with inputs $i, j, k \in \{0,1\}^N$ and $b \in \{0,1\}^3$. We write $i = (i_1, i_2)$, where i_1 is of length $N - n$ and i_2 is of length n, and likewise for j and k. Let this circuit act as follows:

   - Compute the $\vee$ of the bits in i_1, j_1, and k_1. Output 0 if this is 1.
   - Otherwise, output $\mathcal{C}_{\mathrm{dec}}(i_2, j_2, k_2, b_1, b_2, b_3)$.

   As defined, this circuit has size $s + 3(N - n) + 2 \leq 4N =: S$, and we will pad it with additional gates in a trivial manner so that it has exactly S gates. It can be checked that it succinctly represents the same 3Sat formula as $\mathcal{C}_{\mathrm{dec}}$.

We set $\mathrm{PadC}(\mathcal{C}_{\mathrm{inst}}) := \mathcal{C}_{\mathrm{pad}}$, $\mathrm{PadN}(\mathcal{C}_{\mathrm{inst}}) := N$, and $\mathrm{PadS}(\mathcal{C}_{\mathrm{inst}}) := 4 \cdot N$. We note that given $\mathcal{C}_{\mathrm{inst}}$, the value of N is efficiently computable.

Introspective Formula Game



Flip an unbiased coin $\boldsymbol{b} \sim \{0,1\}$.

- Player $\boldsymbol{b}$: Give $(Z,\ Z,\ Z,\ Z,$ "formula"); receive $\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3, (\boldsymbol{b}, \boldsymbol{w})$ and $\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \boldsymbol{\nu}_3$ and $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{M'}$.

Compute $\mathrm{sat}_{\psi,\boldsymbol{\nu}}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{w})$ and $\mathrm{zero}_{H,\boldsymbol{\mu}}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{w})$. Accept if they are equal.

Figure 11: The game $\mathscr{G}_{\mathrm{IntroForm}}(\mathcal{C}_{\mathrm{inst}}, h, q, m)$.

Notation 15.2. In the classical case (Section 11), we have a fixed proof which contains fixed functions which may or may not be low-degree. In the quantum case, however, we are dealing not with a fixed proof but an interactive prover, and the formula prover may not respond based on fixed functions (their responses might be randomized, for example). To account for this, we modify the definitions of sat and zero as follows. First, we recall the notation $g_\psi := g_{\psi,n,t_1,t_2}$ (Definition 11.3).

- Given $\nu_1, \nu_2, \nu_3 \in \mathbb{F}_q$, define

$$\mathrm{sat}_{\psi,\nu}(x, b, w) := g_\psi(x, b, w) \cdot (\nu_1 - b_1)(\nu_2 - b_2)(\nu_3 - b_3).$$

- Given $\mu_1, \ldots, \mu_{m'} \in \mathbb{F}_q$, define

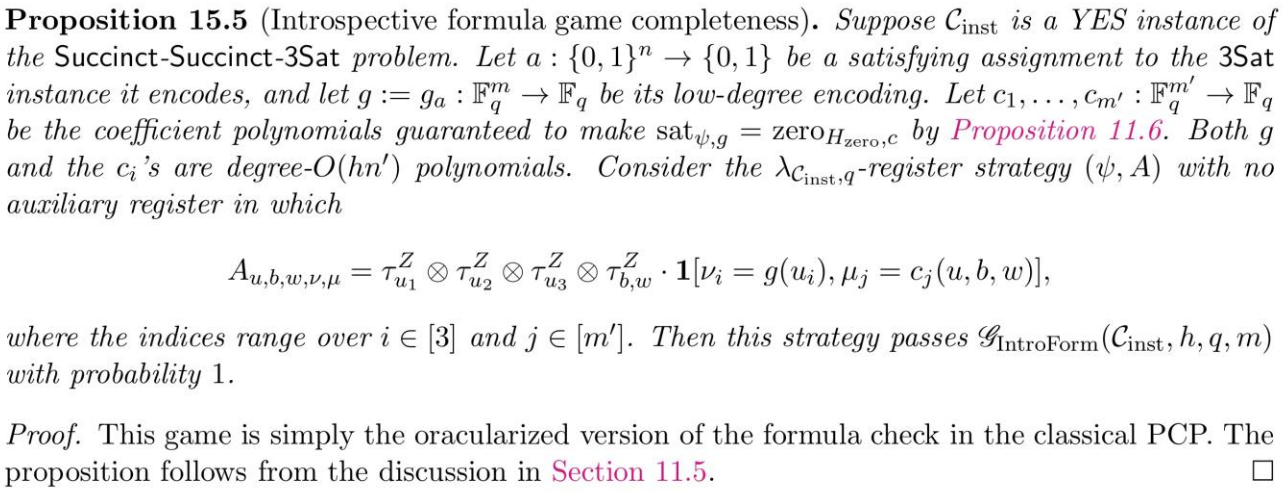
$$\mathrm{zero}_{H,\mu}(x) = \sum_{i=1}^{m'} \mathrm{zero}_{(H_{\mathrm{zero}})_i}(x_i) \cdot \mu_i,$$

  where by definition $(H_{\mathrm{zero}})_i = H$ for $i \in [3m]$ and $(H_{\mathrm{zero}})_i = \{0,1\}$ otherwise.

We note that if there is a function g such that $\nu_i = g(x_i)$, then $\mathrm{sat}_{\psi,\nu} = \mathrm{sat}_{\psi,g}$. Similarly, if there are functions $c_1, \ldots, c_{m'}$ such that $\mu_i = c_i(x)$, then $\mathrm{zero}_{H,\mu} = \mathrm{zero}_{H,c}$.

Now we state the introspective formula game.

Introspective Formula game

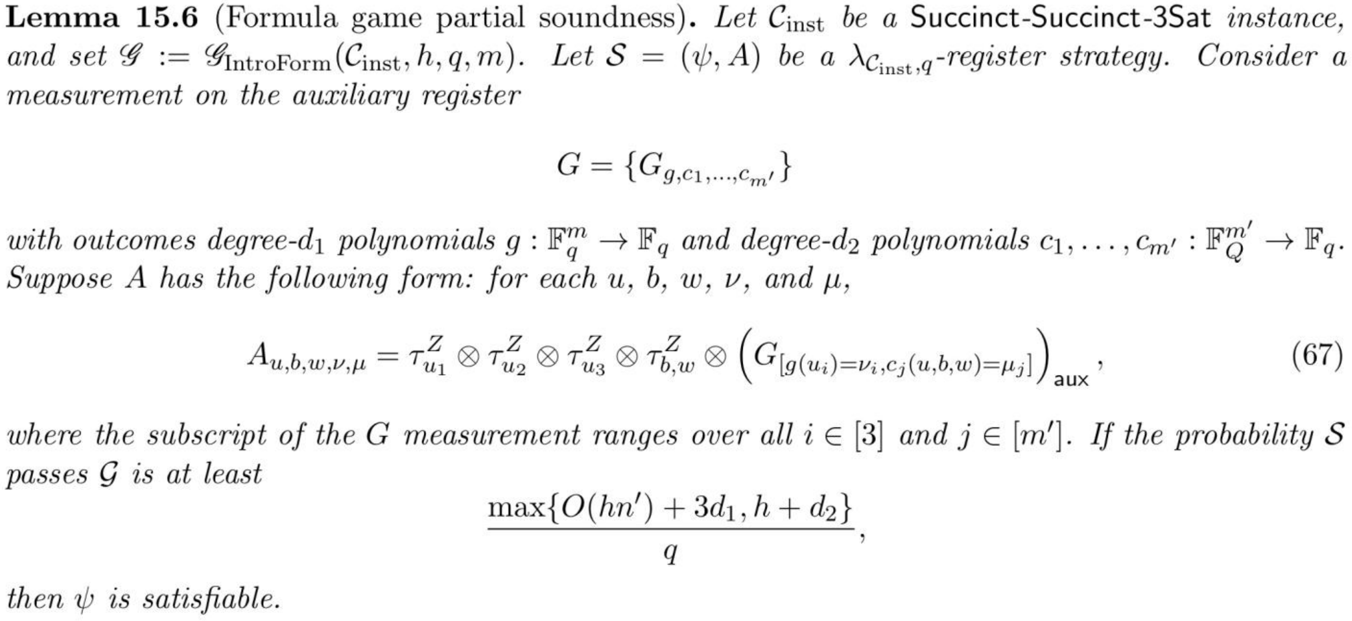


Proposition 15.5 (Introspective formula game completeness)**.** *Suppose $\mathcal{C}_{\text{inst}}$ is a YES instance of the* Succinct-Succinct-3Sat *problem. Let $a : \{0,1\}^n \to \{0,1\}$ be a satisfying assignment to the* 3Sat *instance it encodes, and let $g := g_a : \mathbb{F}_q^m \to \mathbb{F}_q$ be its low-degree encoding. Let $c_1, \ldots, c_{m'} : \mathbb{F}_q^{m'} \to \mathbb{F}_q$ be the coefficient polynomials guaranteed to make $\text{sat}_{\psi,g} = \text{zero}_{H_{\text{zero}},c}$ by Proposition 11.6. Both g and the c_i's are degree-$O(hn')$ polynomials. Consider the $\lambda_{\mathcal{C}_{\text{inst}},q}$-register strategy (ψ, A) with no auxiliary register in which*

$$A_{u,b,w,\nu,\mu} = \tau^Z_{u_1} \otimes \tau^Z_{u_2} \otimes \tau^Z_{u_3} \otimes \tau^Z_{b,w} \cdot \mathbf{1}[\nu_i = g(u_i), \mu_j = c_j(u,b,w)],$$

where the indices range over $i \in [3]$ and $j \in [m']$. Then this strategy passes $\mathscr{G}_{\text{IntroForm}}(\mathcal{C}_{\text{inst}}, h, q, m)$ with probability 1.

Proof. This game is simply the oracularized version of the formula check in the classical PCP. The proposition follows from the discussion in Section 11.5. □



Lemma 15.6 (Formula game partial soundness)**.** *Let $\mathcal{C}_{\text{inst}}$ be a* Succinct-Succinct-3Sat *instance, and set $\mathscr{G} := \mathscr{G}_{\text{IntroForm}}(\mathcal{C}_{\text{inst}}, h, q, m)$. Let $\mathcal{S} = (\psi, A)$ be a $\lambda_{\mathcal{C}_{\text{inst}},q}$-register strategy. Consider a measurement on the auxiliary register*

$$G = \{G_{g,c_1,\ldots,c_{m'}}\}$$

with outcomes degree-d_1 polynomials $g : \mathbb{F}_q^m \to \mathbb{F}_q$ and degree-d_2 polynomials $c_1, \ldots, c_{m'} : \mathbb{F}_Q^{m'} \to \mathbb{F}_q$. Suppose A has the following form: for each u, b, w, ν, and μ,

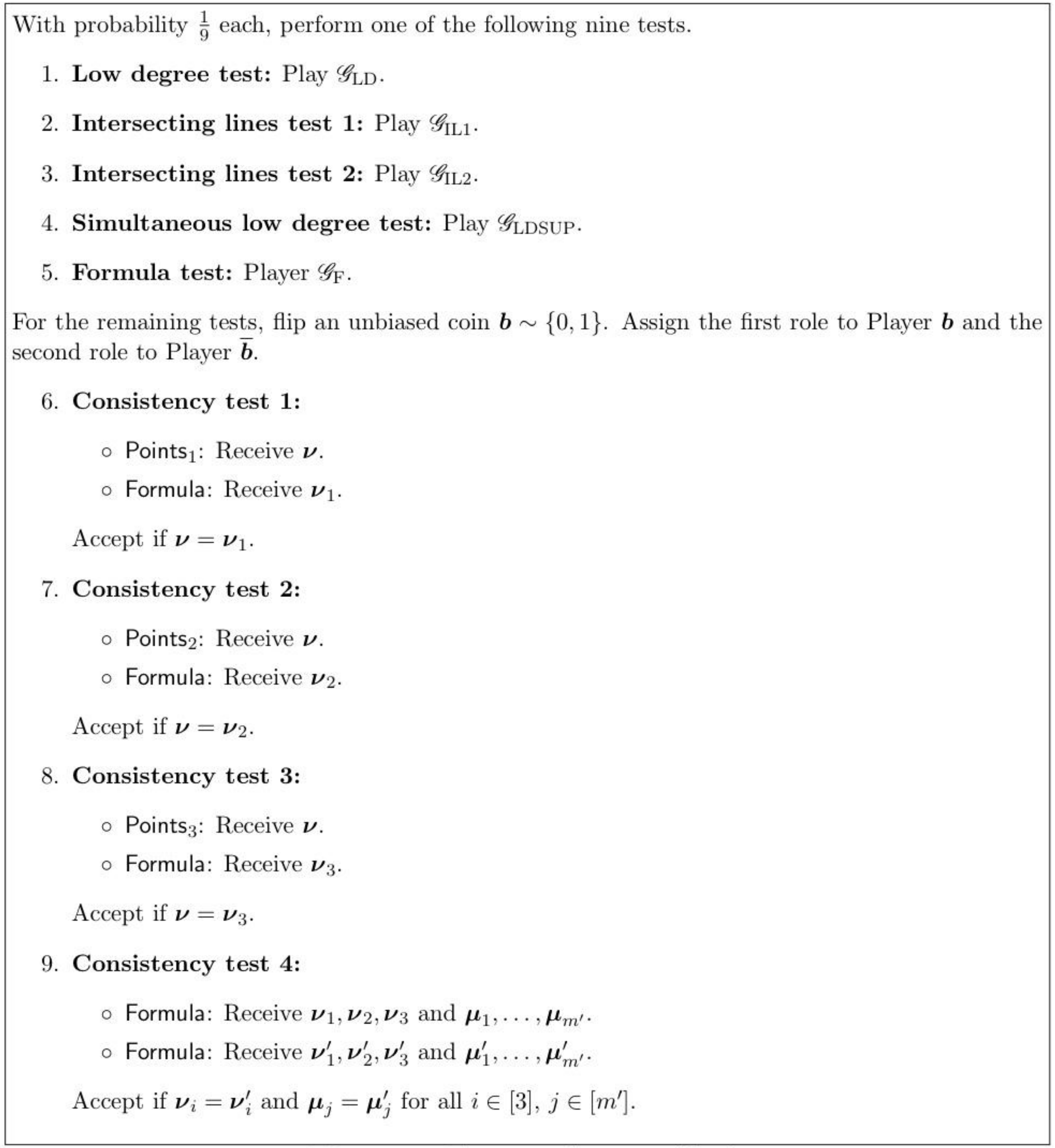
$$A_{u,b,w,\nu,\mu} = \tau^Z_{u_1} \otimes \tau^Z_{u_2} \otimes \tau^Z_{u_3} \otimes \tau^Z_{b,w} \otimes \left(G_{[g(u_i)=\nu_i, c_j(u,b,w)=\mu_j]}\right)_{\mathsf{aux}}, \tag{67}$$

where the subscript of the G measurement ranges over all $i \in [3]$ and $j \in [m']$. If the probability $\mathcal{S}$ passes $\mathcal{G}$ is at least

$$\frac{\max\{O(hn') + 3d_1, h + d_2\}}{q},$$

then ψ is satisfiable.

IntroNEEXP



With probability $\frac{1}{9}$ each, perform one of the following nine tests.

1. **Low degree test:** Play $\mathscr{G}_{\mathrm{LD}}$.
2. **Intersecting lines test 1:** Play $\mathscr{G}_{\mathrm{IL1}}$.
3. **Intersecting lines test 2:** Play $\mathscr{G}_{\mathrm{IL2}}$.
4. **Simultaneous low degree test:** Play $\mathscr{G}_{\mathrm{LDSUP}}$.
5. **Formula test:** Player $\mathscr{G}_{\mathrm{F}}$.

For the remaining tests, flip an unbiased coin $\boldsymbol{b} \sim \{0,1\}$. Assign the first role to Player $\boldsymbol{b}$ and the second role to Player $\overline{\boldsymbol{b}}$.

6. **Consistency test 1:**
   - Points_1: Receive $\boldsymbol{\nu}$.
   - $\mathsf{Formula}$: Receive $\boldsymbol{\nu}_1$.

   Accept if $\boldsymbol{\nu} = \boldsymbol{\nu}_1$.
7. **Consistency test 2:**
   - Points_2: Receive $\boldsymbol{\nu}$.
   - $\mathsf{Formula}$: Receive $\boldsymbol{\nu}_2$.

   Accept if $\boldsymbol{\nu} = \boldsymbol{\nu}_2$.
8. **Consistency test 3:**
   - Points_3: Receive $\boldsymbol{\nu}$.
   - $\mathsf{Formula}$: Receive $\boldsymbol{\nu}_3$.

   Accept if $\boldsymbol{\nu} = \boldsymbol{\nu}_3$.
9. **Consistency test 4:**
   - $\mathsf{Formula}$: Receive $\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \boldsymbol{\nu}_3$ and $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{m'}$.
   - $\mathsf{Formula}$: Receive $\boldsymbol{\nu}'_1, \boldsymbol{\nu}'_2, \boldsymbol{\nu}'_3$ and $\boldsymbol{\mu}'_1, \ldots, \boldsymbol{\mu}'_{m'}$.

   Accept if $\boldsymbol{\nu}_i = \boldsymbol{\nu}'_i$ and $\boldsymbol{\mu}_j = \boldsymbol{\mu}'_j$ for all $i \in [3]$, $j \in [m']$.

Figure 12: The game $\mathscr{G}_{\mathrm{Intro}\mathsf{NEEXP}}(\mathcal{C}_{\mathrm{inst}})$.

Theorem 15.8. *Let $\mathcal{C}_{\text{inst}}$ be a size-(s_{inst}) $\mathsf{Succinct\text{-}Succinct\text{-}3Sat}$ instance. Let q be a sufficiently large $\text{poly}(n)$ and $\epsilon > 0$ a sufficiently small constant such that Equation (72) is at least $\frac{1}{2}$ and Equation (73) is less than $\frac{1}{2}$. Write $\mathscr{G} := \mathscr{G}_{\text{Intro}\mathsf{NEEXP}}(\mathcal{C}_{\text{inst}})$.*

- ***Completeness:** Suppose $\mathcal{C}_{\text{inst}}$ encodes a satisfiable formula. Then there is a value-1 λ-register strategy for $\mathscr{G}$ with no auxiliary register.*
- ***Soundness:** If there is a λ-register strategy for $\mathscr{G}$ with value at least $1-\epsilon$, then $\mathcal{C}_{\text{inst}}$ encodes a satisfiable formula.*

Furthermore,

$$\mathsf{Q\text{-}length}(\mathscr{G}) = O(1), \quad \mathsf{A\text{-}length}(\mathscr{G}) = \text{poly}(2^{s_{\text{inst}}}),$$
$$\mathsf{Q\text{-}time}(\mathscr{G}) = O(1), \quad \mathsf{A\text{-}time}(\mathscr{G}) = \text{poly}(2^{s_{\text{inst}}}).$$

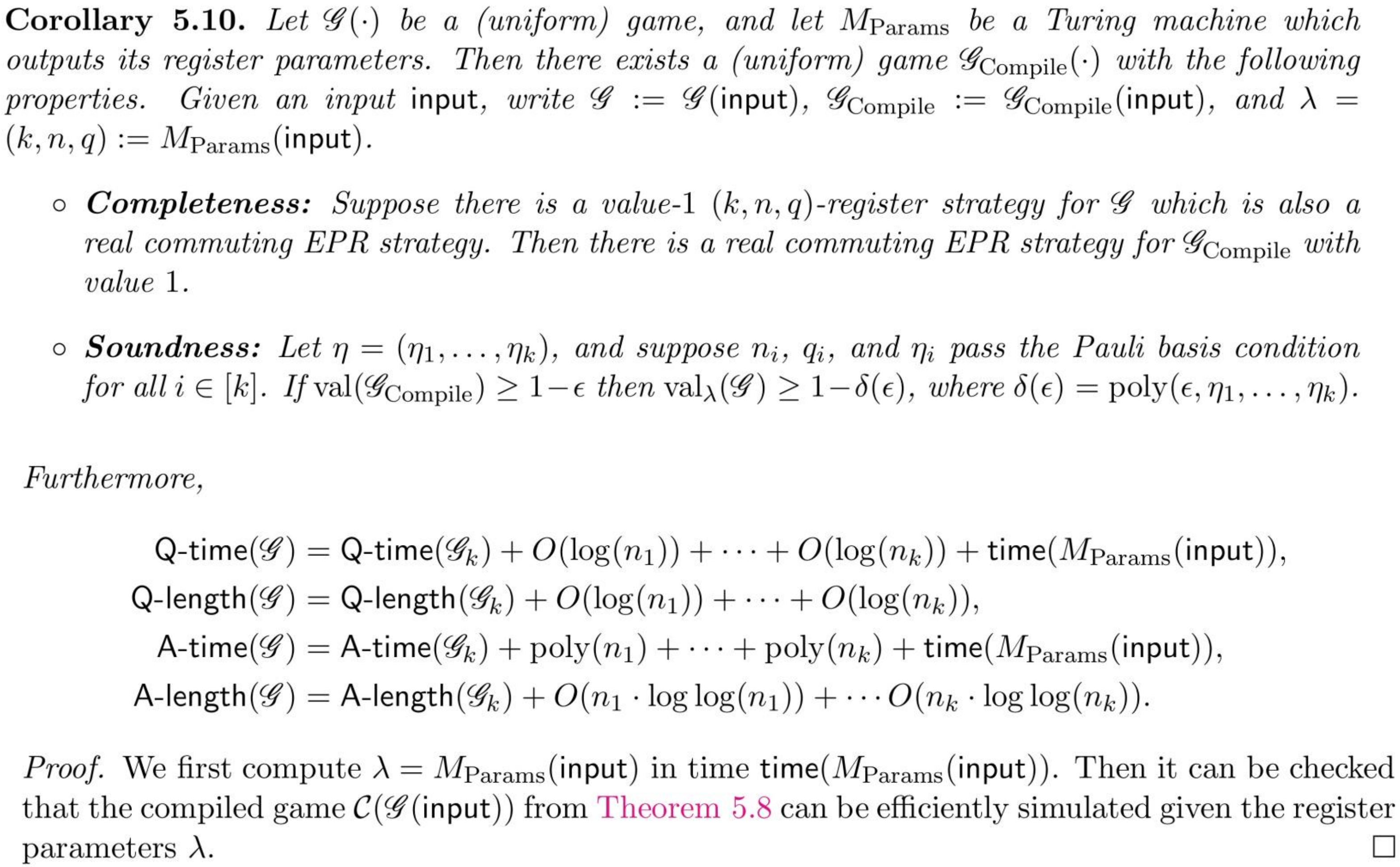
Corollary 15.9. *There is an absolute constant $\epsilon > 0$ such that the following is true. Let $\mathcal{C}_{\text{inst}}$ be a size-(s_{inst}) $\mathsf{Succinct\text{-}Succinct\text{-}3Sat}$ instance. Then there exists a game $\mathscr{G} := \mathscr{G}_{\text{Intro}\mathsf{NEEXP}}(\mathcal{C}_{\text{inst}})$ with the following properties.*

- ***Completeness:** Suppose $\mathcal{C}_{\text{inst}}$ encodes a satisfiable formula. Then there is a value-1 real commuting EPR strategy for $\mathscr{G}$.*
- ***Soundness:** If there is a strategy for $\mathscr{G}$ with value at least $1-\epsilon$, then $\mathcal{C}_{\text{inst}}$ encodes a satisfiable formula.*

Furthermore,

$$\mathsf{Q\text{-}length}(\mathscr{G}) = O(s_{\text{inst}}), \quad \mathsf{A\text{-}length}(\mathscr{G}) = \text{poly}(2^{s_{\text{Inst}}}),$$
$$\mathsf{Q\text{-}time}(\mathscr{G}) = O(s_{\text{inst}}), \quad \mathsf{A\text{-}time}(\mathscr{G}) = \text{poly}(2^{s_{\text{inst}}}).$$

Complier



Corollary 5.10. *Let $\mathscr{G}(\cdot)$ be a (uniform) game, and let M_{Params} be a Turing machine which outputs its register parameters. Then there exists a (uniform) game $\mathscr{G}_{\text{Compile}}(\cdot)$ with the following properties. Given an input* input, *write $\mathscr{G} := \mathscr{G}(\mathsf{input})$, $\mathscr{G}_{\text{Compile}} := \mathscr{G}_{\text{Compile}}(\mathsf{input})$, and $\lambda = (k, n, q) := M_{\text{Params}}(\mathsf{input})$.*

- ***Completeness:*** *Suppose there is a value-1 (k, n, q)-register strategy for $\mathscr{G}$ which is also a real commuting EPR strategy. Then there is a real commuting EPR strategy for $\mathscr{G}_{\text{Compile}}$ with value 1.*

- ***Soundness:*** *Let $\eta = (\eta_1, \ldots, \eta_k)$, and suppose n_i, q_i, and η_i pass the Pauli basis condition for all $i \in [k]$. If $\mathrm{val}(\mathscr{G}_{\text{Compile}}) \geq 1 - \epsilon$ then $\mathrm{val}_\lambda(\mathscr{G}) \geq 1 - \delta(\epsilon)$, where $\delta(\epsilon) = \mathrm{poly}(\epsilon, \eta_1, \ldots, \eta_k)$.*

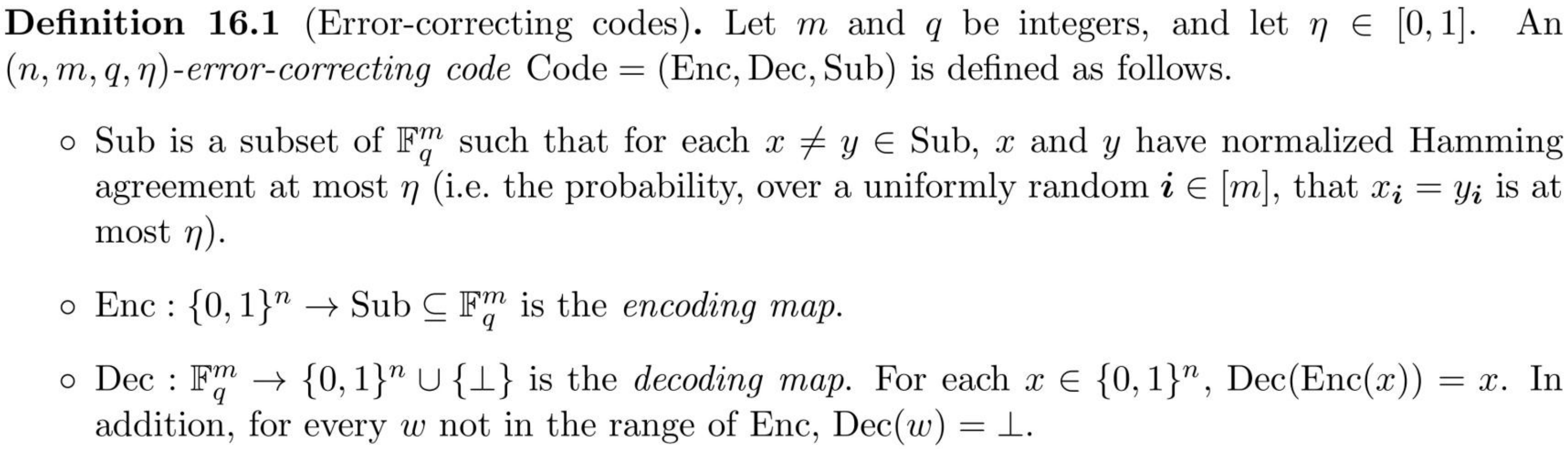
Furthermore,

$$\begin{aligned}
\mathsf{Q\text{-}time}(\mathscr{G}) &= \mathsf{Q\text{-}time}(\mathscr{G}_k) + O(\log(n_1)) + \cdots + O(\log(n_k)) + \mathsf{time}(M_{\text{Params}}(\mathsf{input})),\\
\mathsf{Q\text{-}length}(\mathscr{G}) &= \mathsf{Q\text{-}length}(\mathscr{G}_k) + O(\log(n_1)) + \cdots + O(\log(n_k)),\\
\mathsf{A\text{-}time}(\mathscr{G}) &= \mathsf{A\text{-}time}(\mathscr{G}_k) + \mathrm{poly}(n_1) + \cdots + \mathrm{poly}(n_k) + \mathsf{time}(M_{\text{Params}}(\mathsf{input})),\\
\mathsf{A\text{-}length}(\mathscr{G}) &= \mathsf{A\text{-}length}(\mathscr{G}_k) + O(n_1 \cdot \log\log(n_1)) + \cdots O(n_k \cdot \log\log(n_k)).
\end{aligned}$$

Proof. We first compute $\lambda = M_{\text{Params}}(\mathsf{input})$ in time $\mathsf{time}(M_{\text{Params}}(\mathsf{input}))$. Then it can be checked that the compiled game $\mathcal{C}(\mathscr{G}(\mathsf{input}))$ from Theorem 5.8 can be efficiently simulated given the register parameters λ. □

3. Answer Reduction

Error Correcting Codes



Definition 16.1 (Error-correcting codes)**.** Let m and q be integers, and let $\eta \in [0,1]$. An (n,m,q,η)-*error-correcting code* $\mathrm{Code} = (\mathrm{Enc}, \mathrm{Dec}, \mathrm{Sub})$ is defined as follows.

- Sub is a subset of $\mathbb{F}_q^m$ such that for each $x \neq y \in \mathrm{Sub}$, x and y have normalized Hamming agreement at most η (i.e. the probability, over a uniformly random $\boldsymbol{i} \in [m]$, that $x_{\boldsymbol{i}} = y_{\boldsymbol{i}}$ is at most η).
- $\mathrm{Enc} : \{0,1\}^n \to \mathrm{Sub} \subseteq \mathbb{F}_q^m$ is the *encoding map*.
- $\mathrm{Dec} : \mathbb{F}_q^m \to \{0,1\}^n \cup \{\perp\}$ is the *decoding map*. For each $x \in \{0,1\}^n$, $\mathrm{Dec}(\mathrm{Enc}(x)) = x$. In addition, for every w not in the range of Enc, $\mathrm{Dec}(w) = \perp$.

Theorem 16.6. *Consider low-degree parameters* $\mathsf{params} = (n, q, h, H, m, \mathcal{S}, \pi)$. *Set* $d = m(h-1)$. *Set* $m' = q^m$. *We will identify strings in* $\mathbb{F}_q^{m'}$ *with functions* $g : \mathbb{F}_q^m \to \mathbb{F}_q$. *Given* $a \in \{0,1\}^n$, *define* $\mathrm{Enc}(a) = g_a$ *and* $\mathrm{Dec}(g_a) = a$. *For all other* $g : \mathbb{F}_q^m \to \mathbb{F}_q$ *(i.e. those which are not the low-degree encoding of a string* a*), define* $\mathrm{Dec}(g) = \perp$. *Finally, define* Sub *to be the set of degree* d *polynomials* $g : \mathbb{F}_q^m \to \mathbb{F}_q$. *Then* $\mathrm{Code} = (\mathrm{Enc}, \mathrm{Dec}, \mathrm{Sub})$ *is an* $(n, m', q, d/q)$*-error-correcting code.*

Furthermore, there exists a constant $c > 0$ *and a function* $\delta(\epsilon) = \mathrm{poly}(\epsilon, dm/q^c)$ *such that the following holds. Let* k *be an integer. Then* $\mathscr{G}_{\mathrm{LDsubset}} := \mathscr{G}_{\mathrm{LDsubset}}(m, q, d, \cdot)$ *is a* k*-subset tester for* LDCode *with robustness* $\delta(\epsilon)$.

Finally,

$$\mathsf{Q\text{-}time}(\mathscr{G}_{\mathrm{LDsubset}}) = \mathrm{poly}(m, k, \log q), \quad \mathsf{A\text{-}time}(\mathscr{G}_{\mathrm{LDsubset}}) = \mathrm{poly}(m, d^k, \log q),$$

$$\mathsf{Q\text{-}length}(\mathscr{G}_{\mathrm{LDsubset}}) = O(km \log q), \quad \mathsf{A\text{-}length}(\mathscr{G}_{\mathrm{LDsubset}}) = O(d^k \log(q)).$$

With probability $\frac{1}{2}$ each, perform one of the following two tests.

1. **Low-degree:** Perform $\mathscr{G}_{\mathrm{Surface}}(m, d, q, 2)$.
2. **Cross-check:** Flip an unbiased coin $\boldsymbol{b} \sim \{0,1\}$. Let $\boldsymbol{s}$ be a uniformly random subspace of dimension $k+1$ containing the points in F. With probability $\frac{1}{2}$ each:

   (a) Let $\boldsymbol{w}$ be a uniformly random point in $\boldsymbol{s}$. Distribute the questions as follows:

   - Player $\boldsymbol{b}$: give $\boldsymbol{w}$; receive a value $\boldsymbol{y} \in \mathbb{F}_q$.
   - Player $\overline{\boldsymbol{b}}$: give $\boldsymbol{s}$; receive a degree-d polynomial $\boldsymbol{g} : \boldsymbol{s} \to \mathbb{F}_q$.

   Accept if $\boldsymbol{g}(\boldsymbol{w}) = \boldsymbol{y}$.

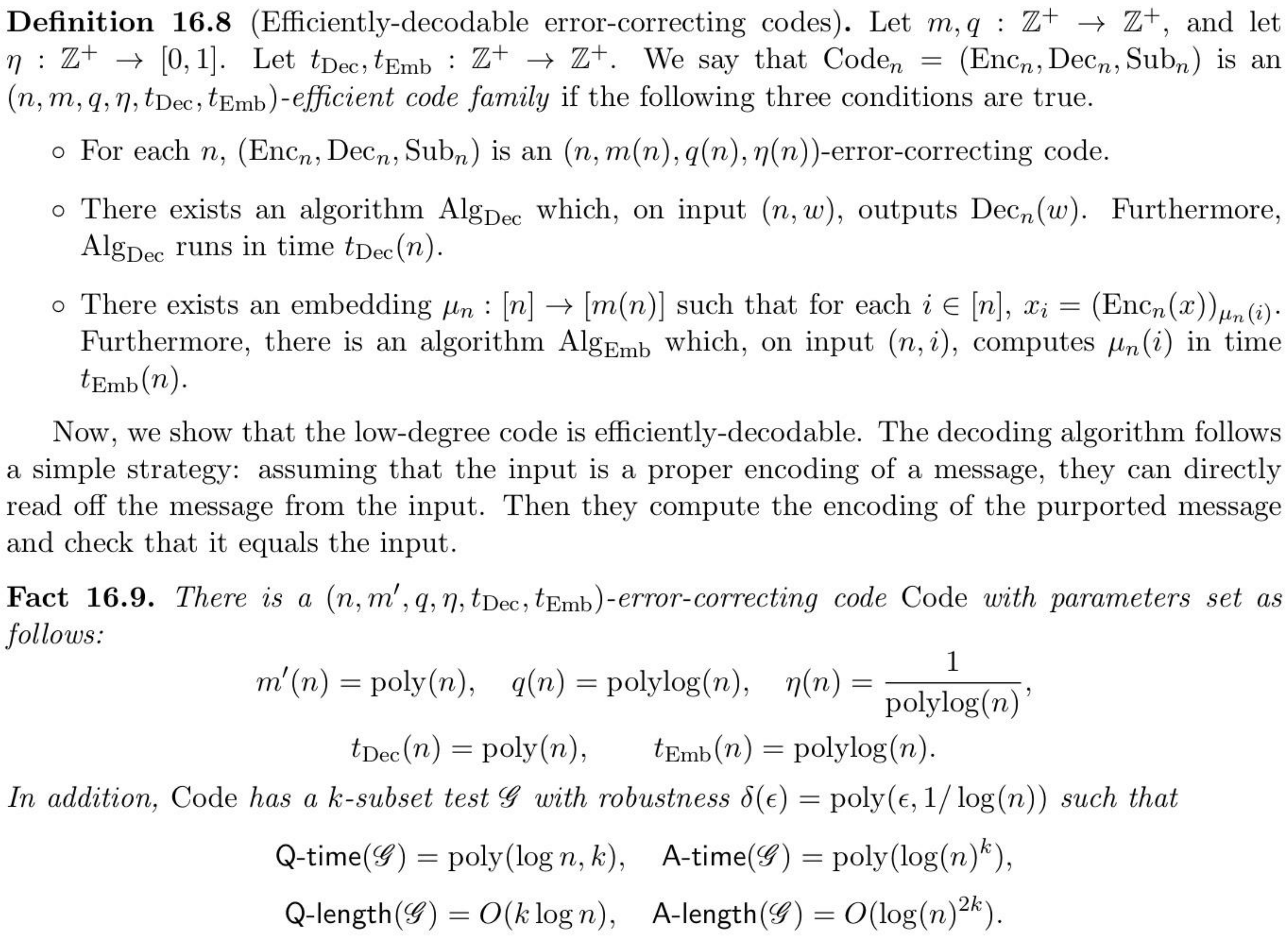
   (b) Distribute the questions as follows:

   - Player $\boldsymbol{b}$: give $\boldsymbol{s}$; receive a degree-d polynomial $\boldsymbol{g} : \boldsymbol{s} \to \mathbb{F}_q$.
   - Player $\overline{\boldsymbol{b}}$: give F; receive a function $\boldsymbol{f} : \boldsymbol{F} \to \mathbb{F}_q$.

   Accept if $\boldsymbol{g}|_F = \boldsymbol{f}$.

Figure 13: The game $\mathscr{G}_{\mathrm{LDsubset}}(m, q, d, F)$.

Efficiently Decodable Codes



Definition 16.8 (Efficiently-decodable error-correcting codes)**.** Let $m, q : \mathbb{Z}^+ \to \mathbb{Z}^+$, and let $\eta : \mathbb{Z}^+ \to [0,1]$. Let $t_{\mathrm{Dec}}, t_{\mathrm{Emb}} : \mathbb{Z}^+ \to \mathbb{Z}^+$. We say that $\mathrm{Code}_n = (\mathrm{Enc}_n, \mathrm{Dec}_n, \mathrm{Sub}_n)$ is an $(n, m, q, \eta, t_{\mathrm{Dec}}, t_{\mathrm{Emb}})$*-efficient code family* if the following three conditions are true.

- For each n, $(\mathrm{Enc}_n, \mathrm{Dec}_n, \mathrm{Sub}_n)$ is an $(n, m(n), q(n), \eta(n))$-error-correcting code.
- There exists an algorithm $\mathrm{Alg}_{\mathrm{Dec}}$ which, on input (n, w), outputs $\mathrm{Dec}_n(w)$. Furthermore, $\mathrm{Alg}_{\mathrm{Dec}}$ runs in time $t_{\mathrm{Dec}}(n)$.
- There exists an embedding $\mu_n : [n] \to [m(n)]$ such that for each $i \in [n]$, $x_i = (\mathrm{Enc}_n(x))_{\mu_n(i)}$. Furthermore, there is an algorithm $\mathrm{Alg}_{\mathrm{Emb}}$ which, on input (n, i), computes $\mu_n(i)$ in time $t_{\mathrm{Emb}}(n)$.

Now, we show that the low-degree code is efficiently-decodable. The decoding algorithm follows a simple strategy: assuming that the input is a proper encoding of a message, they can directly read off the message from the input. Then they compute the encoding of the purported message and check that it equals the input.

Fact 16.9. *There is a* $(n, m', q, \eta, t_{\mathrm{Dec}}, t_{\mathrm{Emb}})$*-error-correcting code* Code *with parameters set as follows:*

$$m'(n) = \mathrm{poly}(n), \quad q(n) = \mathrm{polylog}(n), \quad \eta(n) = \frac{1}{\mathrm{polylog}(n)},$$

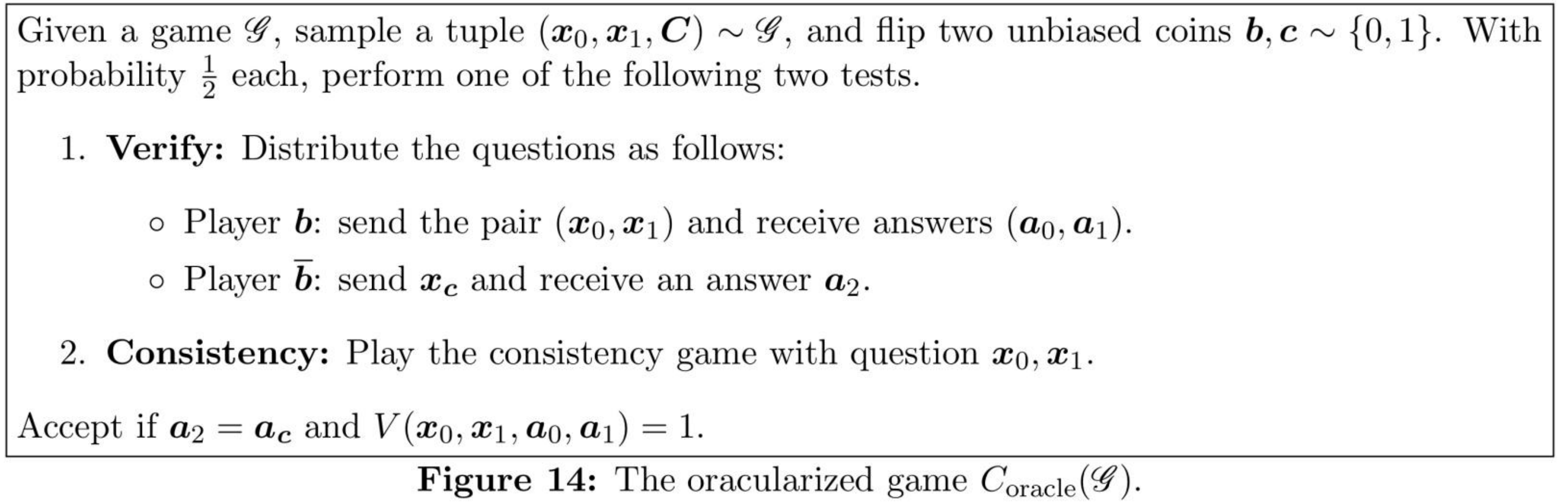
$$t_{\mathrm{Dec}}(n) = \mathrm{poly}(n), \qquad t_{\mathrm{Emb}}(n) = \mathrm{polylog}(n).$$

In addition, Code *has a k-subset test $\mathscr{G}$ with robustness* $\delta(\epsilon) = \mathrm{poly}(\epsilon, 1/\log(n))$ *such that*

$$\textsf{Q-time}(\mathscr{G}) = \mathrm{poly}(\log n, k), \quad \textsf{A-time}(\mathscr{G}) = \mathrm{poly}(\log(n)^k),$$

$$\textsf{Q-length}(\mathscr{G}) = O(k \log n), \quad \textsf{A-length}(\mathscr{G}) = O(\log(n)^{2k}).$$

Oracularization

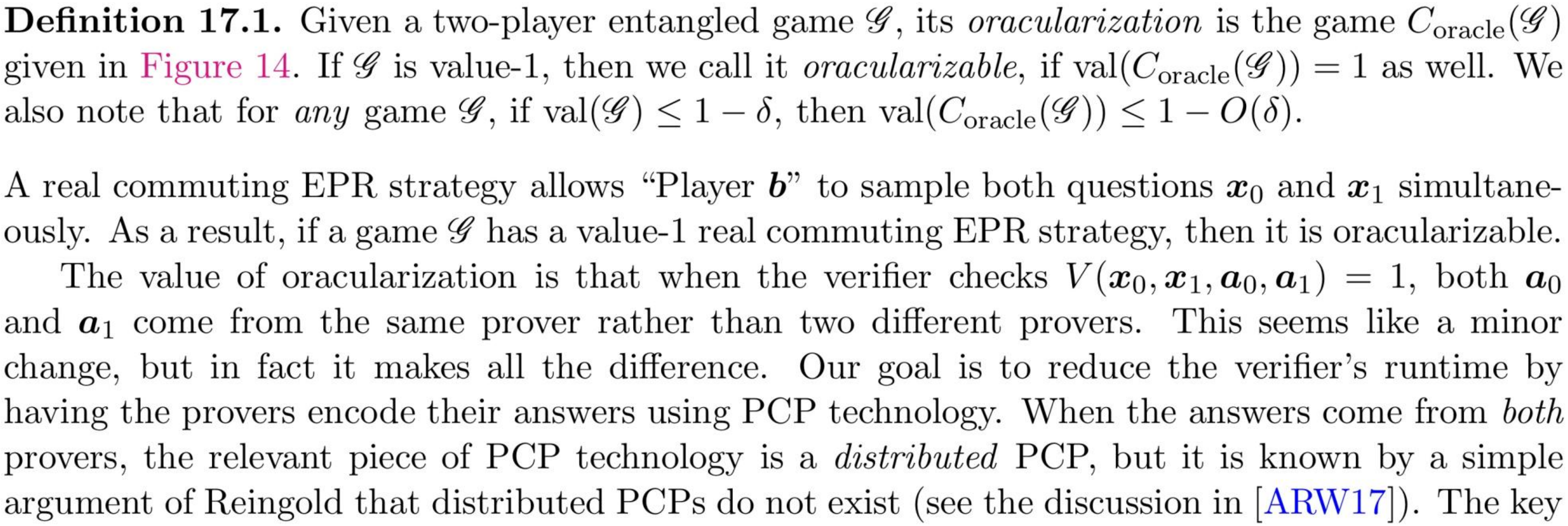


Given a game $\mathscr{G}$, sample a tuple $(\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{C}) \sim \mathscr{G}$, and flip two unbiased coins $\boldsymbol{b}, \boldsymbol{c} \sim \{0,1\}$. With probability $\frac{1}{2}$ each, perform one of the following two tests.

1. **Verify:** Distribute the questions as follows:
   - ○ Player $\boldsymbol{b}$: send the pair $(\boldsymbol{x}_0, \boldsymbol{x}_1)$ and receive answers $(\boldsymbol{a}_0, \boldsymbol{a}_1)$.
   - ○ Player $\overline{\boldsymbol{b}}$: send $\boldsymbol{x}_{\boldsymbol{c}}$ and receive an answer $\boldsymbol{a}_2$.
2. **Consistency:** Play the consistency game with question $\boldsymbol{x}_0, \boldsymbol{x}_1$.

Accept if $\boldsymbol{a}_2 = \boldsymbol{a}_{\boldsymbol{c}}$ and $V(\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{a}_0, \boldsymbol{a}_1) = 1$.

Figure 14: The oracularized game $C_{\text{oracle}}(\mathscr{G})$.

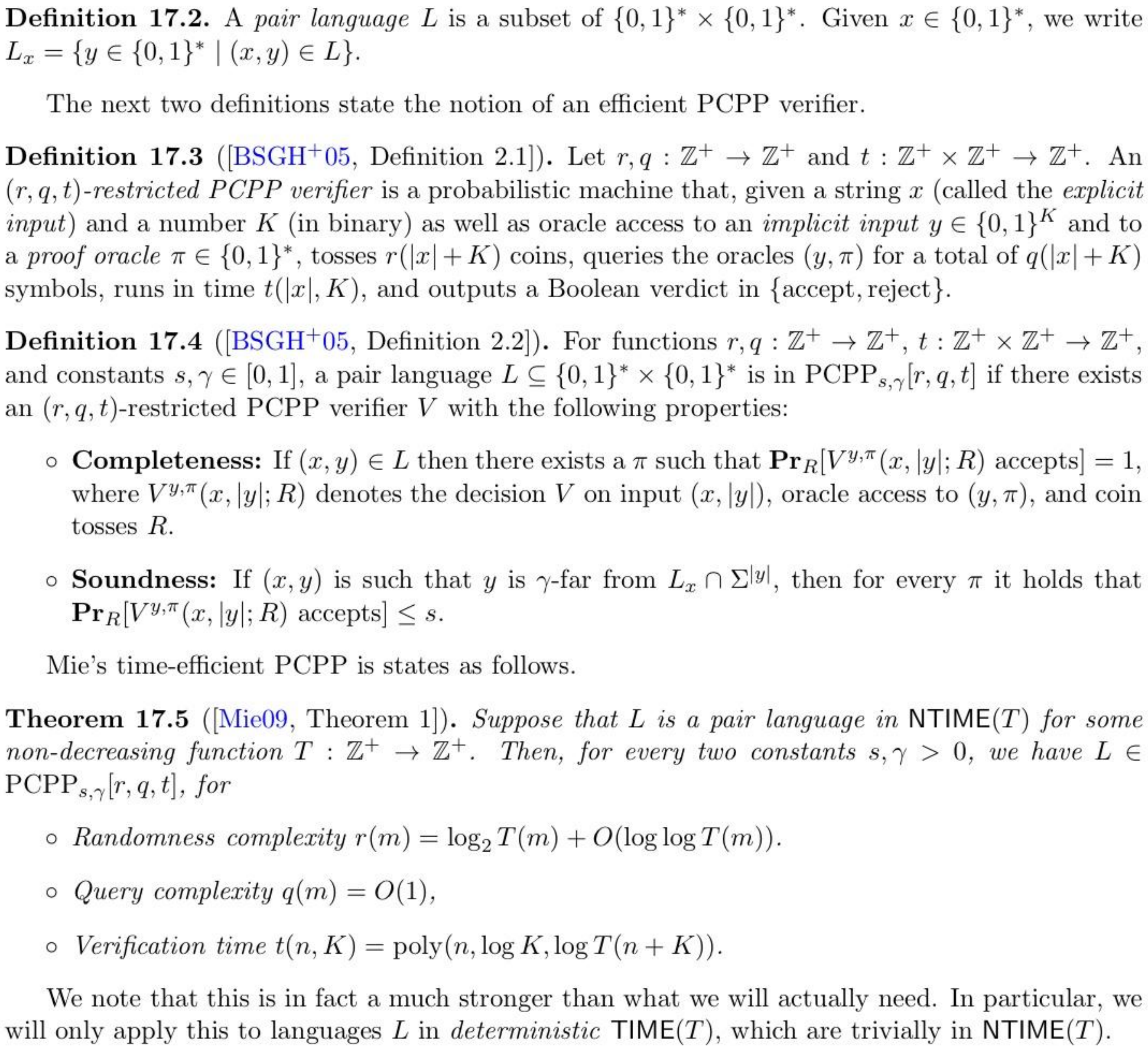


Definition 17.1. Given a two-player entangled game $\mathscr{G}$, its *oracularization* is the game $C_{\text{oracle}}(\mathscr{G})$ given in Figure 14. If $\mathscr{G}$ is value-1, then we call it *oracularizable*, if $\text{val}(C_{\text{oracle}}(\mathscr{G})) = 1$ as well. We also note that for *any* game $\mathscr{G}$, if $\text{val}(\mathscr{G}) \leq 1 - \delta$, then $\text{val}(C_{\text{oracle}}(\mathscr{G})) \leq 1 - O(\delta)$.

A real commuting EPR strategy allows "Player $\boldsymbol{b}$" to sample both questions $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ simultaneously. As a result, if a game $\mathscr{G}$ has a value-1 real commuting EPR strategy, then it is oracularizable.

The value of oracularization is that when the verifier checks $V(\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{a}_0, \boldsymbol{a}_1) = 1$, both $\boldsymbol{a}_0$ and $\boldsymbol{a}_1$ come from the same prover rather than two different provers. This seems like a minor change, but in fact it makes all the difference. Our goal is to reduce the verifier's runtime by having the provers encode their answers using PCP technology. When the answers come from *both* provers, the relevant piece of PCP technology is a *distributed* PCP, but it is known by a simple argument of Reingold that distributed PCPs do not exist (see the discussion in [ARW17]). The key

PCPP



Definition 17.2. A *pair language* L is a subset of $\{0,1\}^* \times \{0,1\}^*$. Given $x \in \{0,1\}^*$, we write $L_x = \{y \in \{0,1\}^* \mid (x,y) \in L\}$.

The next two definitions state the notion of an efficient PCPP verifier.

Definition 17.3 ([BSGH$^+$05, Definition 2.1])**.** Let $r, q : \mathbb{Z}^+ \to \mathbb{Z}^+$ and $t : \mathbb{Z}^+ \times \mathbb{Z}^+ \to \mathbb{Z}^+$. An (r,q,t)*-restricted PCPP verifier* is a probabilistic machine that, given a string x (called the *explicit input*) and a number K (in binary) as well as oracle access to an *implicit input* $y \in \{0,1\}^K$ and to a *proof oracle* $\pi \in \{0,1\}^*$, tosses $r(|x|+K)$ coins, queries the oracles (y,π) for a total of $q(|x|+K)$ symbols, runs in time $t(|x|,K)$, and outputs a Boolean verdict in $\{\text{accept}, \text{reject}\}$.

Definition 17.4 ([BSGH$^+$05, Definition 2.2])**.** For functions $r, q : \mathbb{Z}^+ \to \mathbb{Z}^+$, $t : \mathbb{Z}^+ \times \mathbb{Z}^+ \to \mathbb{Z}^+$, and constants $s, \gamma \in [0,1]$, a pair language $L \subseteq \{0,1\}^* \times \{0,1\}^*$ is in $\mathrm{PCPP}_{s,\gamma}[r,q,t]$ if there exists an (r,q,t)-restricted PCPP verifier V with the following properties:

- **Completeness:** If $(x,y) \in L$ then there exists a π such that $\mathbf{Pr}_R[V^{y,\pi}(x,|y|;R) \text{ accepts}] = 1$, where $V^{y,\pi}(x,|y|;R)$ denotes the decision V on input $(x,|y|)$, oracle access to (y,π), and coin tosses R.

- **Soundness:** If (x,y) is such that y is γ-far from $L_x \cap \Sigma^{|y|}$, then for every π it holds that $\mathbf{Pr}_R[V^{y,\pi}(x,|y|;R) \text{ accepts}] \le s$.

Mie's time-efficient PCPP is states as follows.

Theorem 17.5 ([Mie09, Theorem 1])**.** *Suppose that L is a pair language in $\mathsf{NTIME}(T)$ for some non-decreasing function $T : \mathbb{Z}^+ \to \mathbb{Z}^+$. Then, for every two constants $s, \gamma > 0$, we have $L \in \mathrm{PCPP}_{s,\gamma}[r,q,t]$, for*

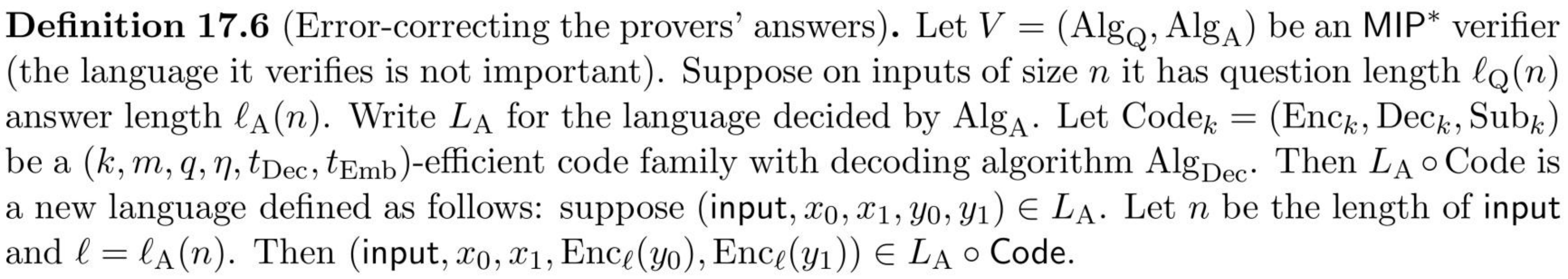
- *Randomness complexity $r(m) = \log_2 T(m) + O(\log\log T(m))$.*

- *Query complexity $q(m) = O(1)$,*

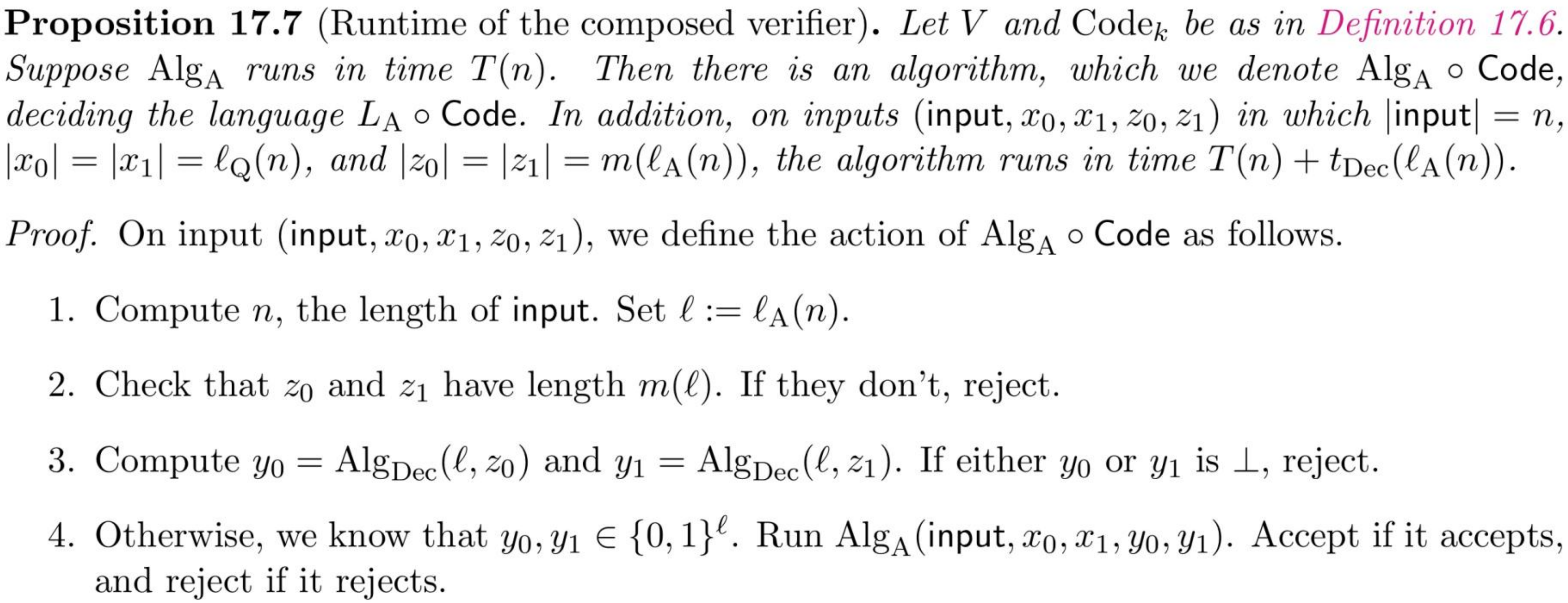
- *Verification time $t(n,K) = \mathrm{poly}(n, \log K, \log T(n+K))$.*

We note that this is in fact a much stronger than what we will actually need. In particular, we will only apply this to languages L in *deterministic* $\mathsf{TIME}(T)$, which are trivially in $\mathsf{NTIME}(T)$.

Composing with error correcting code



Definition 17.6 (Error-correcting the provers' answers)**.** Let $V = (\mathrm{Alg}_{\mathrm{Q}}, \mathrm{Alg}_{\mathrm{A}})$ be an MIP^* verifier (the language it verifies is not important). Suppose on inputs of size n it has question length $\ell_{\mathrm{Q}}(n)$ answer length $\ell_{\mathrm{A}}(n)$. Write L_{A} for the language decided by $\mathrm{Alg}_{\mathrm{A}}$. Let $\mathrm{Code}_k = (\mathrm{Enc}_k, \mathrm{Dec}_k, \mathrm{Sub}_k)$ be a $(k, m, q, \eta, t_{\mathrm{Dec}}, t_{\mathrm{Emb}})$-efficient code family with decoding algorithm $\mathrm{Alg}_{\mathrm{Dec}}$. Then $L_{\mathrm{A}} \circ \mathrm{Code}$ is a new language defined as follows: suppose $(\mathsf{input}, x_0, x_1, y_0, y_1) \in L_{\mathrm{A}}$. Let n be the length of input and $\ell = \ell_{\mathrm{A}}(n)$. Then $(\mathsf{input}, x_0, x_1, \mathrm{Enc}_\ell(y_0), \mathrm{Enc}_\ell(y_1)) \in L_{\mathrm{A}} \circ \mathsf{Code}$.



Proposition 17.7 (Runtime of the composed verifier)**.** *Let V and Code_k be as in Definition 17.6. Suppose $\mathrm{Alg}_{\mathrm{A}}$ runs in time $T(n)$. Then there is an algorithm, which we denote $\mathrm{Alg}_{\mathrm{A}} \circ \mathsf{Code}$, deciding the language $L_{\mathrm{A}} \circ \mathsf{Code}$. In addition, on inputs $(\mathsf{input}, x_0, x_1, z_0, z_1)$ in which $|\mathsf{input}| = n$, $|x_0| = |x_1| = \ell_{\mathrm{Q}}(n)$, and $|z_0| = |z_1| = m(\ell_{\mathrm{A}}(n))$, the algorithm runs in time $T(n) + t_{\mathrm{Dec}}(\ell_{\mathrm{A}}(n))$.*

Proof. On input $(\mathsf{input}, x_0, x_1, z_0, z_1)$, we define the action of $\mathrm{Alg}_{\mathrm{A}} \circ \mathsf{Code}$ as follows.

1. Compute n, the length of input. Set $\ell := \ell_{\mathrm{A}}(n)$.
2. Check that z_0 and z_1 have length $m(\ell)$. If they don't, reject.
3. Compute $y_0 = \mathrm{Alg}_{\mathrm{Dec}}(\ell, z_0)$ and $y_1 = \mathrm{Alg}_{\mathrm{Dec}}(\ell, z_1)$. If either y_0 or y_1 is $\perp$, reject.
4. Otherwise, we know that $y_0, y_1 \in \{0,1\}^\ell$. Run $\mathrm{Alg}_{\mathrm{A}}(\mathsf{input}, x_0, x_1, y_0, y_1)$. Accept if it accepts, and reject if it rejects.

Definition 17.9. We instantiate the answer-reduced MIP^* protocol with the following algorithms and parameters.

- Let $V = (\mathrm{Alg}_{\mathrm{Q}}, \mathrm{Alg}_{\mathrm{A}})$ be an MIP^* verifier for a language L. Write L_{A} for the language decided by $\mathrm{Alg}_{\mathrm{A}}$. Suppose on inputs of size n, the verifier V has question length $\ell_{V,\mathrm{Q}}(n)$, answer length $\ell_{V,\mathrm{A}}(n)$, question time $t_{V,\mathrm{Q}}(n)$, and answer time $t_{V,\mathrm{A}}(n)$.

- Let $\mathrm{Code}_k = (\mathrm{Enc}_k, \mathrm{Dec}_k, \mathrm{Sub}_k)$ be a $(k, m, q, \eta, t_{\mathrm{Dec}}, t_{\mathrm{Emb}})$-efficient code family with decoding algorithm $\mathrm{Alg}_{\mathrm{Dec}}$ and embedding μ_k.

- Let $\mathscr{G}_k$ be a game which tests for Code_k with robustness $\chi_k(\epsilon)$. Suppose it has question length $\ell_{\mathscr{G},\mathrm{Q}}(k)$, answer length $\ell_{\mathscr{G},\mathrm{A}}(k)$, question time $t_{\mathscr{G},\mathrm{Q}}(k)$, and answer time $t_{\mathscr{G},\mathrm{A}}(k)$.

- Let $s, \delta > 0$ be constants, and let V_{PCPP} be the PCPP verifier for the language $L_{\mathrm{A}} \circ \mathsf{Code}$ guaranteed by Theorem 17.5 with these parameters. Suppose on inputs of size n it has proof length $\ell_\pi(n)$. By Proposition 17.7, $L_{\mathrm{A}} \circ \mathrm{Code}$ is in time $t_{\mathrm{compose}}(n) = t_{V,\mathrm{A}}(n) + t_{\mathrm{Dec}}(\ell_{V,\mathrm{A}}(n))$. We can therefore write V_{PCPP}'s verification time as

  $$t_{\mathrm{PCPP}}(n) = \mathrm{poly}(n + \ell_{V,\mathrm{Q}}(n), \log(m(\ell_{V,\mathrm{A}}(n))), \log(t_{\mathrm{compose}}(n))).$$

  Finally, $\ell_\pi(n) = t_{\mathrm{compose}}(n) \cdot \mathrm{polylog}(t_{\mathrm{compose}}(n))$.

Write $\ell_1 := \ell_{V,\mathrm{A}}(n)$ and $\ell_2 := \ell_\pi(n)$. Then the *answer reduction game* $\mathscr{G}_{\mathrm{answer}}(\mathsf{input}; V, \mathrm{Code}, \mathscr{G}, s, \delta)$ is given in Figure 15. We write V_{answer} for the corresponding verifier.

Theorem 17.10. *Suppose V, Code, $\mathscr{G}$, and V_{PCPP} are as in Definition 17.9. Suppose s, γ are chosen to be constants such that $\eta(k) \geq 2\gamma$ for all k. Suppose further that V has the following property: for any input in L, the provers have a real commuting EPR strategy with value 1. Then V_{answer} is also an MIP^* verifier for L with the following two conditions:*

- *(Completeness) If $\mathsf{input} \in L$, then there is a value-1 strategy.*

- *(Soundness) Given input, suppose there is a strategy with value $1-\epsilon$. Then there is a strategy for V on input input with value $1 - \delta(\epsilon)$, where $\delta(\epsilon)$ is given by*

  $$\delta(\epsilon) := \mathrm{poly}(\chi_{\ell_1}(\mathrm{poly}(\epsilon)), \chi_{\ell_2}(\mathrm{poly}(\epsilon)), \eta(\ell_1), \eta(\ell_2)).$$

Hence, if we choose our parameters so that $1 - \delta(\epsilon)$ is greater than the soundness of V, this implies that V_{answer} is an MIP^* verifier for L with soundness $1 - \epsilon$.

감사합니다